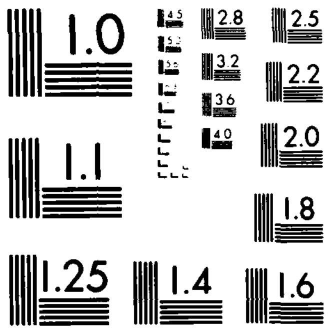
AD-A149 012 EFFICIENT ALGORITHM FOR FUZZY LINEAR PROGRAMMING WITH MULTIPLE OBJECTIVES(U) INFORM G M B H AACHEN (GERMANY F R) R HILLEKAMP ET AL. 01 DEC 84 DAJA45-82-C-0009 1/2

UNCLASSIFIED F/G 12/1 NL




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

(2)

AD



# "EFFICIENT ALGORITHM FOR FUZZY LINEAR PROGRAMMING WITH MULTIPLE OBJECTIVES".

Final Technical Report

by

R. Hillekamp
G.W. Hintz
Dr. B. Werners
P. Zysno
Prof. Dr. H.-J. Zimmermann
(Principal Investigator)

October 1984

United States Army
EUROPEAN RESEARCH OFFICE OF THE U.S. ARMY
London England

CONTRACT NUMBER : DAJA 45 - 82 - C - 0009

Institut für Operations Research und
Management, INFORM GMBH, Aachen

Approved for Public Release; distribution unlimited

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER — | 2. GOVT ACCESSION NO. AD A149 012 | 3. RECIPIENT'S CATALOG NUMBER — |
| 4. TITLE (and Subtitle) Efficient Algorithm For Fuzzy Linear Programming With Multiple Objectives | | 5. TYPE OF REPORT & PERIOD COVERED Final technical |
| | | 6. PERFORMING ORG. REPORT NUMBER - |
| 7. AUTHOR(s) R. Hillekamp, G.W. Hintz, B. Werners, H.-J. Zimmermann, P. Zysno | | 8. CONTRACT OR GRANT NUMBER(s) DAJA 45-82-C-0009 (R&D4135-CC) |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS INFORM Institut für Operations Research und Management, GmbH. Metzgerstr. 1-3 5100 Aachen F.R.G. | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102 A IT 161102BH57-03 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Research, Development and Standardization Group-UK Box 65 FPO, NY 09510 | | 12. REPORT DATE Dec. 1, 1984 |
| | | 13. NUMBER OF PAGES 150 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709 | | 15. SECURITY CLASS. (of this report) unclassified |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, Distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

--

18. SUPPLEMENTARY NOTES

--

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Decision Support System, Fuzzy Linear Programming, Multi Criteria Optimization, Interactive Decision Making

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

On the basis of fuzzy linear programming an interactive Decision Support System has been developed which can handle mathematical programming problems with many crisp or fuzzy obejctive functions and crisp as well as flexible constraints. Empirically tested connectives and membership functions have been included

Table of contents

## Abstract

Linear Programming as an aid to solve certain types of decision problems with many decision variables and many constraints has proofed its power already in the military as well as in the civilian area. Efficient algorithms have been developed and are available as computer programs.

In two aspects, however, these algorithms have not yet been advanced satisfactorily :

1. Most of the algorithms can only accomodate one decision criterion (one objective function).
2. Objectives and constraints usually have to be formulated "crisply" i.e. the objective normally is to be maximized or minimized and the constraints divide decision alternatives into feasable and nonfeasable ones without taking into consideration that in human decision making there are "grey zones".
   In 1965 L.A. Zadeh (Berkeley/USA) suggested the "Theory of Fuzzy Sets" to cope with vagueness of reality when modelling it as mathematical models. 1972 fuzzy decisions were defined by Bellman and Zadeh as the intersection of fuzzy sets representing not crisply defined objectives and constraints which are not of the yes-no or black-white type.

   In the meantime fuzzy linear programming has been introduced and the application of fuzzy linear programming to problems with multiple objectives and fuzzy and crisp constraints was suggested. Here "fuzzy" can either be interpreted as "not crisp" or as "flexibility providing"

   To make these promising approaches useful for the solution of large problems of this type the existing models have been advanced in the following directions:

1. Realistic and empirically tested membership functions are integrated into fuzzy programming models.
2. Adequate connectives for human decision making have been included into these models.
3. An interactive decision support system for decisions with multiple (fuzzy) objectives and crisp and fuzzy constraints has been developed which is user-oriented enough to be accepted by decision makers.

Keywords :

Decision Support System, Fuzzy Linear Programming, Multi Criteria Optimization, Interactive Decision Making.

## I. Statement of the problem

Linear Programming as an aid to solve certain types of decision problems with many decision variables and many constraints has proofed its power already in the military as well as in the civilian area. Efficient algorithms have been developed and are available as computer programs.

In two aspects, however, these algorithms have not yet been advanced satisfactorily:

1. Most of the algorithms can only accommodate one decision criterion (one objective function).
   It has become a generally accepted fact that in many instances many decision criteria need to be considered. Two types of approaches have been suggested so far to cope with this problem: Global Methods (Goal Programming, Utility Models) and Interactive Models.

   The former generally demand more information from the decision-maker than he is able to provide, the latter are generally too inefficient computationally to be used for large problems.

2. Objectives and constraints usually have to be formulated "crisply", i.e. the objective normally is to be maximized or minimized and the constraints divide decision alternatives into feasable and nonfeasable ones without taking into consideration that in human decision making there are "grey zones". In other words a model which its based on traditional, dichotomous, two valued logic cannot model human decision problems properly since reality is not dichotomous but rather of the "more or less tvpe". Thus, problems are frequently modelled in a way such that they are computationally solvable but not such that they describe the real problem properly.

In 1965 L.A. Zadeh (Berkeley/USA) suggested the "Theory of Fuzzy Sets" to cope with vagueness of reality when modelling it as mathematical models. 1972 fuzzy decisions were defined by Bellman and Zadeh as the intersection of fuzzy sets representing not crisply defined objectives and constraints which are not of the yes-no or black-white type.

In the meantime fuzzy linear programming has been introduced and the application of fuzzy linear programming to problems with multiple objectives and fuzzy and crisp constraints was suggested.

To make these promissing approaches useful for the solution of large problems of this type the existing models have been advanced in the following directions:

1. Realistic and empirically tested membership functions are integrated into fuzzy programming models.
2. Adequate connectives for human decision making have been included into these models.
3. An interactive decision support system for decisions with multiple (fuzzy) objectives and crisp and fuzzy constraints has been developed which is user-oriented enough to be accepted by decision makers.
4. This system has to be programmed and tested such that it can also be used for large decision problems.

The project aims to combine and advance the results which have already been achieved by a team of mathematicians, management scientists, computer scientists, and psychologists in Aachen during the last 4 years for the development of a system including the above mentioned four properties.

# II. Basic Theory

## 1. Historical Background

(Basic Theory of Fuzzy Sets)

The use of mathematical models and methods in order to gain more insight into the functioning of complex systems and in order to find optimal solutions to problems has been steadily increasing in the past. Even though considerable successes could be achieved by this approach certain limitations became more and more obvious when moving into the areas of human systems and decision-making where the systems to be modelled are very complex. Two of the major reasons for this are :

1. A major part of classical mathematics is based on "crisp", two valued logic, i.e. assuming that certain facts or relations are either true or not true. In human life i.e. whenever human value judgements play an important role, situations can often not be reduced to that type of structure.
2. As we try to tackle more and more complex systems we find that an adequately detailed mathematical model of the problem situation cannot be constructed without loosing the main advantages of mathematical models.

A person which is faced with a problem of the type described above has essentially 5 possible ways of proceeding:

1. He can request that the poser of the problem formulates his problem in a way suitable for mathematical modelling. (In most cases the problem poser will not be able or willing to do this!)

2. The model builder can try to design a mathematical model which approximates the real problem. This, however, enhances the danger that the model is too much influenced by existing known mathematical methods and models available to the model builder and too little by the problem itself. Thus the modelled problem might be solved but not the real one.

3. All persons concerned might be content with a model, which by use of a living language describes well the problem situation but which is not suitable for mathematical description or solution. Two consequences might result:

   (a) Since our day-to-day languages are not unequivocal the model might be ambiguous, and dangerous misinterpretations might be possible.

   (b) Solutions arrived at from such a model will presumably not be too informative to the decisionmaker or even not helpful at all.

4. Use "subjective probabilities" to express the fuzziness of the respective components of the system i.e. work with an axiomatic system designed for stochastic systems and not for fuzzy systems.

5. The expert might eventually use the terminology of the theory of fuzzy sets to describe the problem situation and fuzzy calculus in order to find optimal solutions to the problem which are more informative than the solutions mentioned under 3..

It is essential to realize the basic difference in vagueness between "Fuzziness" and "Probabilistics". While a statement such as: "The chances of horse A winning the race are .5 and that horse B will win are .4" is probabilistic in nature, the statements: "I like all goodlooking girls" or "We have to achieve satisfactory profits" have a "fuzzy" meaning. The nature of "probabilistic" information is different from the nature of fuzzy information and so are the axiomatic systems for probability theory and the theory of fuzzy sets.

L.A. Zadeh suggested in 1965 (Zadeh 1965) the notion of a fuzzy set and the basic theory of fuzzy sets essentially as the link between vague real phenomena and their adequate mathematical modelling.
He defines a fuzzy set as follows :

Definition: If $X = \{x\}$ is a collection of objects denoted generically by x then a fuzzy set A in X is a set of ordered pairs.

$$(1) \quad A = \{(x, \mu_A(x)) \ , \ x \in X\}$$

$\mu_A(x)$ is called the membership function or grade of membership[1)] of x in A which maps X tothe membership-space M. (When M contains only the two points 0 and 1, A is nonfuzzy and $\mu_A(x)$ is identical to the characteristic function of a nonfuzzy set).
$\mu(\cdot)$ is a function the range of which is a subset of the nonnegative real numbers and has the property that the supremum of this set is finite.

## Decisions in Fuzzy Environments

In conventional nonfuzzy decisionmaking under certainty we are used to thinking of a decision as consisting of

(a) a set of possible activities (decision variables),

(b) a set of constraints limiting the choice between the alternatives (solution space) and

(c) the objective function which assigns a "value" to each result due to a certain choice of activities according to their "desirability". The optimal decision is then the selection of the activity with the highest "desirability" (for instance the alternative which results in minimum cost, maximum profit etc.).

1) also degree of compatibility or degree of truth

In a fuzzy environment this picture of a decision has to be revised: The fuzzy objective function is characterized by its membership function, so are the constraints. Since we want to satisfy (optimize) the objective function as well as the constraints, a decision in a fuzzy environment is defined in analogy to nonfuzzy environments as the selection of activities which simultaneously satisfy objective function(s) <u>and</u> constraints.

Bellman and Zadeh (Bellman, Zadeh 1970) assumed the logical "and" to correspond to the intersection of the sets to be "merged" and therefore defined a "fuzzy decision" as the intersection of fuzzy constraints and fuzzy objective function(s). The relationship between constraints and objective functions in a fuzzy envirement are therefore fully symmetric, i.e. there is no longer a difference between the former and latter. This can be illustrated by using the following example :

Example 1

<u>Objective Function</u>: "x should be substantially larger than 10", characterized by the membership function

$$\mu_O(x) = \begin{cases} 0, & x < 10 \\ (1+(x-10)^{-2})^{-1}, & x \geq 10 . \end{cases}$$

<u>Constraint:</u>

"x should be in the vicinity of 11", characterized by the membership function

$$\mu_C(x) = ((1+(x-11)^4)^{-1}$$

The membership function $\mu_D(x)$ of the decision is then

$$\mu_D(x) = \mu_O(x) \wedge \mu_C(x)$$

$$\mu_D(X) = \begin{cases} \text{Min}\left((1+(x-10)^{-2})^{-1},\ (1+(x-11)^4)^{-1}\right) & \text{for } x \geq 10 \\ 0 & \text{for } x < 10 \end{cases}$$

This relation is depicted in Figure 1.

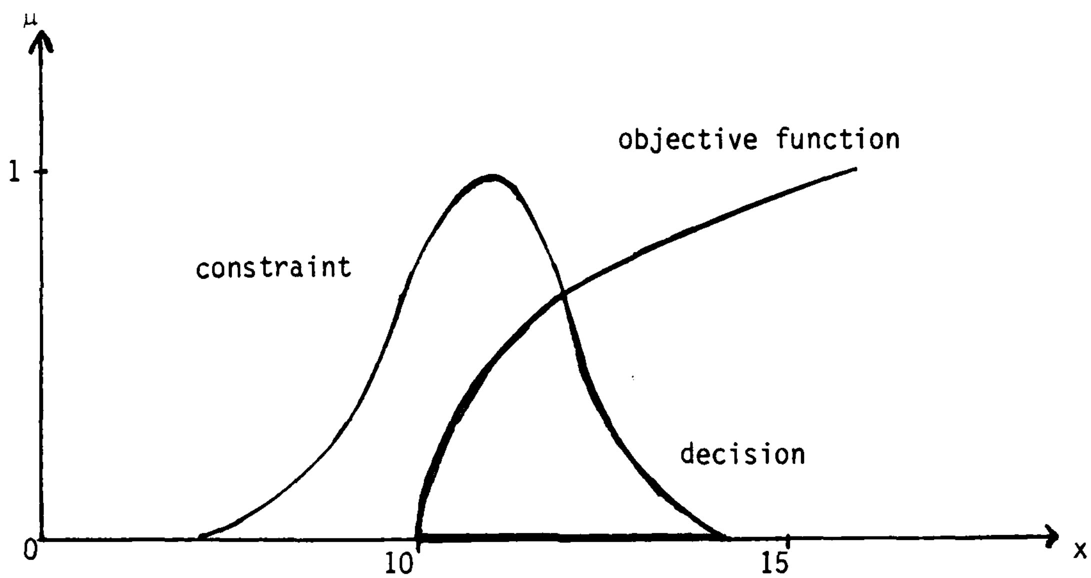



Figure 1

To single out a specific solution from the fuzzy set "decision" it is plausible to select the solution with the highest degree of membership.

## Fuzzy Linear Programming

Linear Programming Problems represent a special but very frequently occurringtype of decision making.

It was therefore quite natural to apply the notion of a fuzzy decision to linear programming. Zimmermann (Zimmermann 1978) suggested a possible way of doing this which can most easily be illustrated by the following example (which stems from a real application) :

Example 2

A company wanted to decide on the size and structure of its truck fleet. Four differently sized trucks ($x_1$ through $x_4$) were considered. The objective was to minimize cost and the constraints were to supply all customers (which had a strong seasonal demand).
That meant: Certain quantities had to be moved (quantity constraint) and a minimum numb of customers per day had to be contacted (routing constraint). Because of other reasons at least 6 of the smallest trucks were wanted in the fleet.
The management wanted to use quantitative analysis and agreed to the following suggested LP-approach (simplified):

$$\begin{array}{ll}
\text{Min} & 41{,}400x_1 + 44{,}300x_2 + 48{,}100x_3 + 49{,}100x_4 \\
\text{s.th.} & 0.84\,x_1 + 1.44\,x_2 + 2.16\,x_3 + 2.40\,x_4 \geq 170.00 \\
 & 16\,x_1 + 16\,x_2 + 16\,x_3 + 16\,x_4 \geq 1300 \\
 & x_1 \geq 6 \\
 & x_1, \ldots x_4 \geq 0
\end{array}$$

With $x_1, \ldots, x_4$ = number of trucks of sizes one through four the solution was $x_1 = 6$, $x_2 = 17.85$, $x_3 = 0$, $x_4 = 58.64$,

$$\text{Min Cost} = 3{,}670{,}850.$$

Since management felt that it was forced into giving precise constraints (because of the model) inspite of the fact that it would rather have given some intervals the following "fuzzy" approach was used:

Starting from the problem

$$\begin{aligned} \text{Min} \quad & Z = cx \\ \text{s.th.} \quad & Ax \le b \\ & x \ge 0 \end{aligned} \tag{1}$$

the adopted "fuzzy" version was

$$\begin{aligned} cx &\lesssim Z \\ Ax &\lesssim b \\ x &\gtrsim 0. \end{aligned} \tag{2}$$

We now define a function f: $R^{m+1} \longrightarrow [0,1]$ such that

$$f(Ax,cx) = \begin{cases} 0 & \text{if } Ax \le b,\ cx \le Z \text{ is strongly violated} \\ & \text{and} \\ 1 & \text{if } Ax \le b \text{ and } cx \le Z \text{ is satisfied.} \end{cases} \tag{3}$$

Using the simplest version of the function f(Ax,cx) we assume it to be linear and the intersection of the (fuzzy) constraints and the (fuzzy) objective function.

Thus $\quad f(Ax,cx) = f(Bx) = \underset{i}{\text{Min}}\ f_i\ ((Bx)_i),\ x \ge 0$

with

$$f_i((Bx)_i) = \begin{cases} 1 & \text{for } (Bx)_i \le b_i \\ 1 - \dfrac{Bx_i - b_i}{d_i} & \text{for } b_i < Bx_i \le b_i + d_i \\ 0 & \text{for } (Bx)_i > b_i + d_i \end{cases} \tag{4}$$

where $d_i$ = subjectively choosen constants of admissable violations of the constraints.

$\underset{i}{\text{Min}}\ f_i\ ((Bx)_i)$ is the membershipfunction of the "fuzzy decision" (5)

and $\underset{x\geq 0}{\text{Max}}\ \underset{i}{\text{Min}}\ f_i\ ((Bx)_i)$ the decision with the highest degree of membership. (6)

Substituting $b_i' = \frac{b_i}{d_i}$

$$Bx_i' = \frac{Bx_i'}{d_i} \qquad \text{componentwise}$$

and simplifying problem (4) by dropping the "1" (which does not change the problem!) we arrive at the following problem:

$$\underset{x\geq 0}{\text{Max}}\ \underset{i}{\text{Min}}\ (b_i' - (Bx)_i') \tag{7}$$

or

$$\underset{x\geq 0}{\text{Max}}\ \mu_D(x)$$

As it is wellknown, problem (7) is equivalent to solving the following LP:

$$\begin{aligned} &\text{Max} && \lambda \\ &\text{s.th.} && \lambda \leq b_i - (Bx)_i, \quad i = 0(1)m \\ &&& x \geq 0 \end{aligned} \tag{8}$$

The optimal solution to ( 8 ) is also the optimal solution to (7).

The following assumptions were made

(1) Total cost should not rise above 4.200,000 (budget limit).

(2) The "unfuzzy" constraints are minimum requirements and management would feel much better if there was some "leeway".

(3) The linear approximations of the membership functions are acceptable.

(4) There are no interdependencies between the constraints.

(5) Weighting of the constraints is taken care of by defining the $d_i$.

(6) The min-operator is the applicable connective.

The theory of fuzzy linear programming has been advanced in the meantime (Hamacher, Leberling, Zimmermann 1978; Rödder, Zimmermann 1980) and fuzzy linear programming has also been applied to a number of problems (for instance Wiedey, Zimmermann 1978; Zimmermann 1980)

## Decision Making in Fuzzy Environments and with Multiple Criteria

Even though Kuhn and Tucker mentioned the "Vector Maximum Problem" already in their publication Nonlinear Programming (Kuhn, Tucker 1951)in 1951, practitioners and the scientific community have only become conscious of the importance of decision-making models which take into consideration several decision criteria since the beginning of the 1970's. Since then a very large number of publications in this area has appeared and the problem can still not be considered as satisfactorily solved. (For a good survey of the State-of-the-Art see for instance (Starr, Zeleny 1977)).

The application of fuzzy linear programming to this problem was first suggested by Zimmermann in 1978 (Zimmermann 1978). This approach seems quite efficient and appropriate.

It lacks, however, in two aspects:

1. It is based on some restrictive assumptions such as linear membershipfunctions, use of the minimum operator.
2. It is a "global model" in the sense that it demands all relevant information from the decision maker before the solution of the problems. (I.e. it assumes that the "pessimistic solution" and the "individual optima" determine the aspiration levels of the decision-maker.) (Thole, Zimmermann, Zysno 1979, Zimmermann, Zysno 1980)

With respect to the first aspect empirical research in Aachen has shown that human decision-makers do not always use the minimum operator but rather operators which alow some degree of compensation. The minimum operator seems to be appropriate for the constraints but not for the combination of the objective functions. The use of other operators (such as suggested in the above references) will result, however, in nonlinear programming models if no appropriate substitutions can be found.

The shape of the membership functions was the subject of an empirical project (financed by the Deutsche Forschungsgemeinschaft) which was completed in 1982.

With respect to the second aspect it seems advisable to develop an interactive model which allows the decision-maker to communicate with the model and to use the "pessimistic" and "optimistic" solutions only as a basis for departure and to approach the "optimal" compromise solution by learning from the model and adapting his aspiration levels accordingly.

## 2. Decision Processes

In section II.1 a decision was interpreted as "finding an optimal solution".

This,however, is not the only possible interpretation. Some authors (and practitioneers) call situations in which "projects" are evaluated and in which "measures of effectivenes " are determined also decisions. If a number of alternatives (alternative actions or projects) are ranked as to their desirability this is also often called a decision. Thus a kind of hierarchy of "decisions" can be formulated :

| degree of Optimization ↓ | solution space → $R^n$ | | |
|---|---|---|---|
| | discrete | cont. | Funct. $S$ |
| Evaluation | 1 | 2 | 3 |
| Ranking | 4 | | 5 |
| Partial Optimization | 6 | 7 | |
| Optimization | 8 | 9 | |

All these (crisp) notions of a decision are non-symmetrical in the sense that the "constraints" or "number of feasable solutions" play a different role than the objective- or utility function.

The concept of a "fuzzy decision" (Bellman, Zadeh 1970) is symmetrical.

This concept was used when designing "fuzzy linear programming" such as described in the last section. This notion is static in the sense that it assumes that "a decision" happens at a point in time. Real decisions, however, are processes which occur over time and which correspond to hierarchies rather than to static models.
It is therefore meaningful to extend the Bellman-Zadeh concept to multi-stage decision processes as follows :

Our paradigm assumes that people either learn or generate "evaluative concepts" or "subjective categories". These terms refer to two sides of one coin : The first refers to the intensional aspects of a set which can be described by a list of attributes and the second stresses the accumulation of objects (extensional aspect of a set). We assume that human beings have such concepts or categories at their disposition and that they can relate them to each other.

Attributes constituting a concept may be interpreted other than psychologically. They can be replaced by any mental information unit, for instance,the status of neutral elements or the adjectives of a language, The relationships may actually be modelled by operators, connectives, rules, or others.

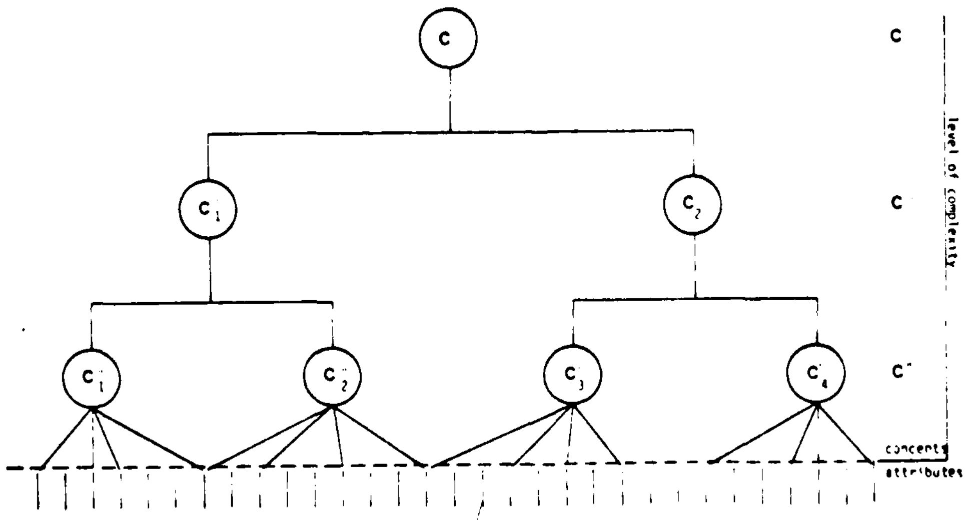



Fig 2 Hierarchy of concepts/categories

For our purposes we will limit considerations to a specific type of amalgamation : We assume a hierarchy of concepts in which there are several levels of complexity. (Fig. 2)
The bottom level contains basic concepts which can stepwise be aggregated until the top concept of the hierarchy is attained. (A more detailed description is given in (Zimmermann, Zysno 1983).

For reasons of practical relevance of the model we shall allow that
(a) the subcategories are of unequal importance for the respective super category.
(b) the description of categories of each level may partly contain the same attributes.

# III. Developments, Results and Conclusions

## III.1 Membership Functions

### Types of functions

Different types of functions can be chosen to express the membership values of elements to a fuzzy set. They mainly differ with respect to their mathematical properties and their empirical fit. We shall first discuss some membership functions introduced in fuzzy sets literature.

The membership function proposed first in connection with mathematical modeling is the linear one (e.g. Zimmermann, 1978). It is uniquely defined by the two values $\underline{c}$ and $\bar{c}$ which have to be provided by a decision maker :

$$\mu(x) = \begin{cases} 0 & x \leq \underline{c} \\ \dfrac{x - \underline{c}}{\bar{c} - \underline{c}} & \underline{c} < x \leq \bar{c} \\ 1 & x \geq \bar{c} \end{cases} \tag{9}$$

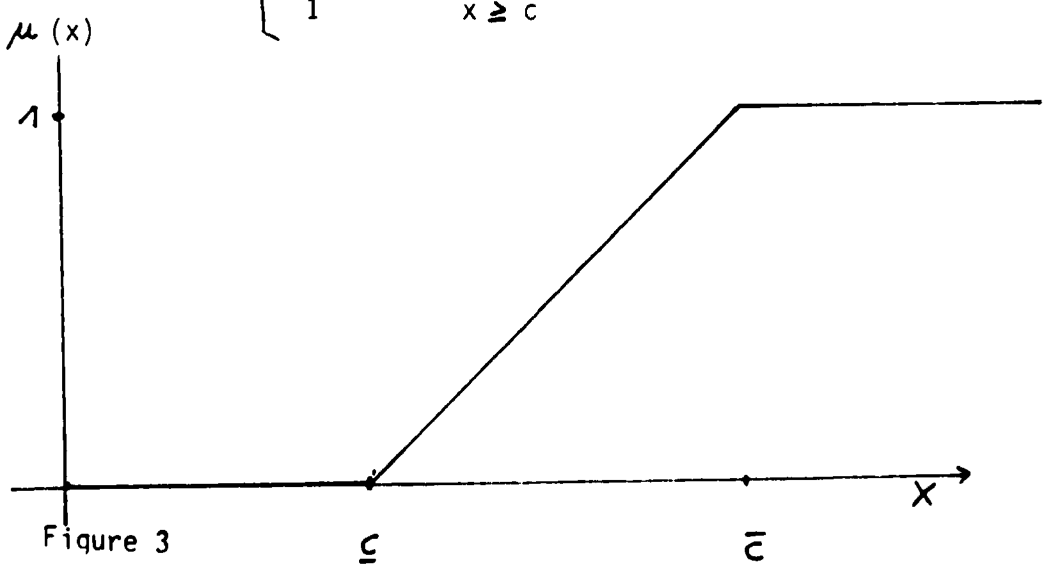



Figure 3

The advantage of the linear function is its good behaviour in linear models. Each function can more or less be approximated by a linear or a piecewise linear function.

Empirical research (e.g. Hersh, Caramazza 1976),however, shows that s-shaped functions model human behaviour much better.
Hence mathematical models of such functions are introduced in fuzzy set literature. For example the logistic function proposed by Zimmermann, Zysno (1984).

$$\mu(x) = \frac{1}{1 \times e^{-a(x-b)}} \tag{10}$$

This function is uniquely defined by the slope a and the inflection point b .

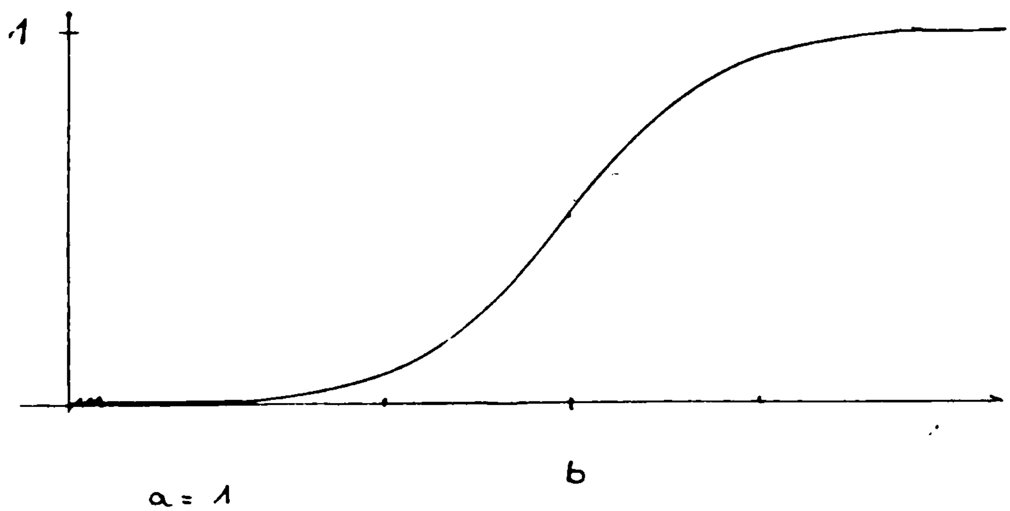



Figure 4

A hyperbolic membership function is proposed by Leberling (1981)

$$\mu_H(x) = \frac{1}{2} \quad \frac{e^{(x-(\bar{c}+\underline{c})/2)\alpha} - e^{-(x-(\bar{c}+\underline{c})/2)\alpha}}{e^{(x-(\bar{c}+\underline{c})/2)\alpha} + e^{-(x-(\bar{c}+\underline{c})/2)\alpha}} \tag{11}$$

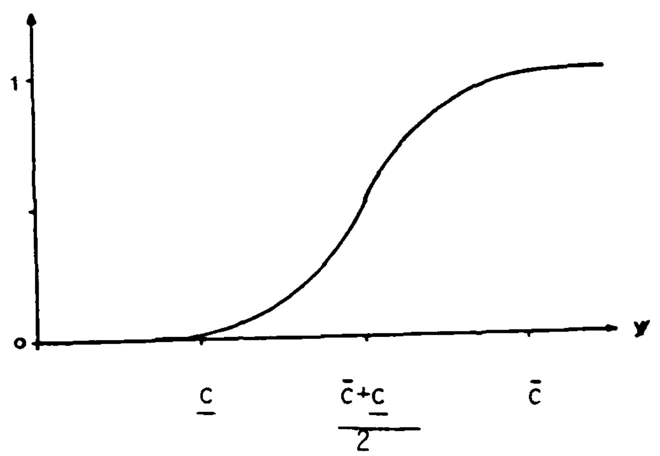



Figure 5

A cubic spline function is deduced by Schwab (1983), here in a cut version :

$$\mu_S(x) = \begin{cases} 1 & \text{für } x \le x_m \\ ax^3 + bx^2 + cx + d & \text{für } x_m \le x \le x_D \\ ex^3 + fx^2 + gx + h & \text{für } x_D \le x \le x_o \\ 0 & \text{für } x_o \le x \end{cases} \tag{12}$$

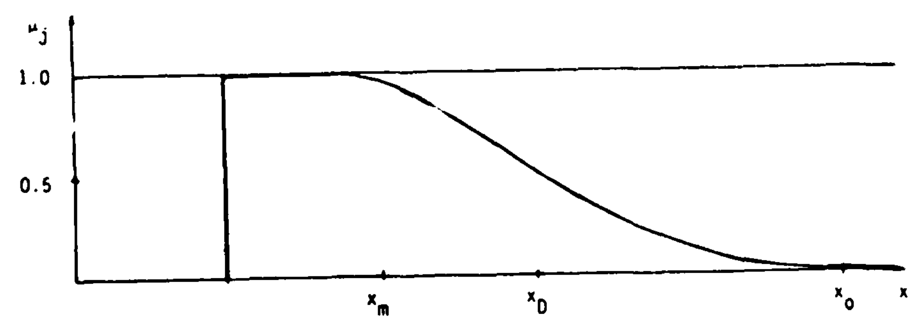



Figure 6

It can be shown that hyperbolic membership function and logistic membership function are isomorphic mathematical models. By choosing appropriate parameters a, b, and $\bar{c}$, $\underline{c}$, $\alpha$ respectively the resulting membership functions are equal.
Determining a cubic spline function many different values, a, b, c, e, f, g, h have to be provided by the decision maker or to be observed empirically. Hence the two most promising membership functions in mathematical modeling are the linear and logistic one which are considered in the following. The linear can be determined easily and can be handled efficiently. The logistic function can also be determined easily and its fit to human behaviour is better.
Some empirical investigations have been performed to further improve the fit.

## Measurement

Measurement means assigning numbers to objects, such that certain relations between numbers reflect analogous relations between objects (Campbell 1938). With other words measurement is the mapping of object relations into numerical relations of the same type.

If it is possible to prove that there is a homomorphic mapping $f : E \to A$ from an empirical relational structure $\langle E, P_1, \ldots, P_n \rangle$ with a set of objects E and n-tuple of relations $P_1$ into a numerical relational structure $\langle N, Q_1, \ldots, Q_n \rangle$ with a set of numbers N and relations $Q_i$, then a scale $\langle\langle E, N, f \rangle\rangle$ exists. By specifying the admissable transformations the grade of uniqueness is determined.

Therefore, measurement starts by formulating the properties of the empirical structure; implicity the intended object space is modelled

on a non-numerical level. Strictly speaking at the very beginning there should be a semantic definition of the central concepts, which would considerably facilitate the consistent use of the relevant principles. This has not yet been possible for the concept of membership. Membership has a clear cut formal definition.
However, apart from first steps by Norwich and Turksen (1984) genuine measurement structures have not yet been developed.

Under these circumstances one could wait and see, until a satisfactory definition is available. However, one should remember that up to the beginning of the 20th century even in the "hard sciences" measures were used without being equipped with adequate measurement theories.
Usually measurement tools were used, which were based on not much more but plausible reasons. Nevertheless, the success of the natural sciences is undoubted. Hence, for the purpose of empirical research it may be tolerable to unse plausible techniques.
As the base variable provides a good deal of control with respect to judgmental errors of the subjects we used direct scaling methods.
This involves less effort in data collection. In order to express this possibly lower level of aspiration we call this scale an "evaluative" scale.

<u>Model</u>

The judgment (valuation) of membership can be regarded as the comparison of object x with a standard (ideal) which results in a distance d(x).
If the object has all the features of the standard the distance shall be zero, if no similarity between standard and object exists, the distance shall be "$\infty$". If the evaluation concept is represented formally by a fuzzy set $\tilde{P} \subseteq X$, then a certain degree of membership $\mu_{\tilde{P}}(x)$ is assigned to each element x. In the following as a matter of convenience we will denote the degree of membership, $\mu_{\tilde{P}}(x)$, simply by $\mu$.

$$\mu = \frac{1}{1+d(x)} \tag{13}$$

Membership is defined as a function of the distance d(x) between a given object x and a standard (ideal). Hence :

$d(x) = 0 \rightarrow \mu = 1$; $d(1) = \infty \rightarrow \mu = 0$. Equation (13) is only a transformation rule from one numerical relative into another : real numbers R are mapped into the interval $[0,1]$.

The distance function now has to be specified. A specific montonic function of the similarity with the ideal could as a first approximation be $d'(x) = 1/x$.

Experience shows, however, that ideals are very rarely ever fully realized. As an aid to determine the relative position very often a context dependent standard b is created.
It facilitates a fast and rough preevaluation such as "rather positive", "rather negative"etc. As another context dependent parameter we can use the evaluation unit a, similar to unit of length such as feet, meters, yards etc. If one realizes furthermore that the relationship between physical unit and perceptions is generally exponential (Helson 1964), then the following distance function seems appropriate :

$$d(x) = \frac{1}{e^{a(x-b)}} \tag{14}$$

Substituting (14) into (13) yields the logistic function

$$\mu = \frac{1}{1+e^{-a(x-b)}} \tag{15}$$

It is S-shaped such as demanded by several authors (Goguen 1969; Zadeh 1971). Formally b is the inflexion point and a is the slope of the function.

From the point of view of linear programming (15) has the additional advantage, that it can easily be linearized by the following transformation :

$$-\ln \frac{1-\mu}{\mu} = \ln \frac{\mu}{1-\mu} = a(x-b). \qquad (16)$$

The parameters a and b will have to be interpreted differently depending on the situation which is modelled. From a linguistic point of view a and b can be considered as semantic parameters.

Since concepts or categories, which are formally represented by sets, are normally linguistically described, the membership function is the formal representation of meaning. The vagueness of the concept is operationalized by the slope a and the identification threshold by b . For managerial terms such as "appropriate dividend" or "good utilization of capacities" a models the slope of the membership function in the tolerance interval und b represents the turning point from rather positive onto rather negative tolerance.

Model (15), however, is still too general to fit subjective models of different persons. Frequently only a certain part of the logistic function is needed to represent a perceived situation. This is also true for measuring devices such as scales, thermometers etc. which are designed for specific measuring areas only.

In order to allow for such a calibration of our model we assume that only a certain interval of the physical scale is mapped into the open interval (0,1) (see figure 7). Whenever stimuli are smaller or equal to the lower bound or larger or equal to the upper bound the grade of membership of 0 or 1 respectively is assigned to them. This is achieved by changing the range by legitimate scale transformations such that the desired interval is mapped into (0,1).

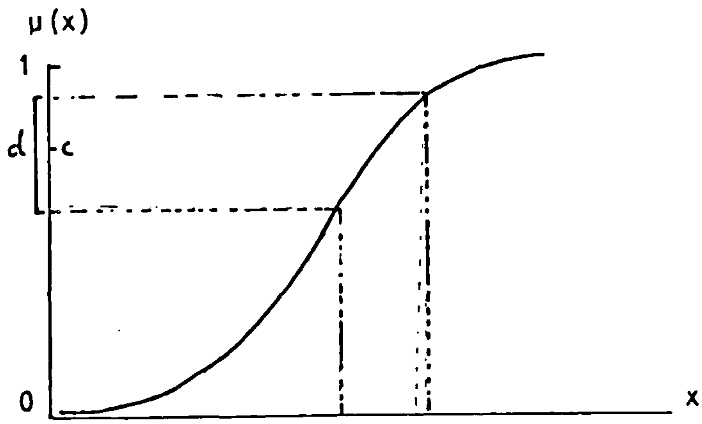



Figure 7 : Calibration of the interval for measurement

The general model of membership (15) is specifical by the two parameters of calibration c and d , c representing the "neutral point" and d the actually used interval.

$$\mu_i = \left\lceil \frac{1}{1 + e^{-a(x - b)}} - c) \frac{1}{d} + \frac{1}{2} \right\rceil_0^1 \qquad (17)$$

$\lceil \; \rceil$ indicates that values outside of the interval ( 0,1 ) have no real meaning. The measurement instrument does not differentiate there.

Hence

$$x < \underline{x} \rightarrow \mu(x) = 0 \qquad (18)$$

$$x > \bar{x} \rightarrow \mu(x) = 1 \qquad (19)$$

The determination of the parameters from an empirical data base does not pose any difficulties in the general model (15).

On the basis of (16) the original membership values are transformed into y-valued :

$$y_i = \ln \frac{\mu}{1-\mu_i} \qquad (20)$$

Beween x and y there exists a linear relationship. The straight line of the model is then defined by the least squares of deviations.

The estimation of the parameters c and d in the extended model still poses some problems. We cannot yet suggest a direct way for a numerically optimal estimation. We can, however, suggest an iterative procedure. We assume that a set of stimuli which is equally spread over the physical continuum was chosen such that the distance between any two of the neighbouring stimuli is constant

$$x_{i+1} - x_i = s \qquad (21)$$

This condition serves as a criterion for precision. If $c$ and $d$ are correctly estimated then those scale values $x_i'$ are reproducable which are invariant with respect to $x_i$ with the exception of the additive and multiplicative constant. This becomes obvious when rewriting (19) as follows :

$$\ln \frac{a(\mu_i - 1/2) + c}{1 - (d(\mu_i - 1/2) + c)} = a(x_i - b) = x_j' \tag{22}$$

Let $s'$ be the distance between the pairs $x_i'$ and $x_{i+1}'$ and $M'$ their mean value. If the estimated values $\hat{d}$ and $\hat{c}$ are equal to their true values then the estimated distance $\hat{s}'$ and the mean $\hat{M}'$ are equal to their respective true values and vice versa :

$$\hat{d} = d \wedge \hat{c} = c \Leftrightarrow \hat{s}' = s \wedge \hat{M}' = M. \tag{23}$$

Our aim is therefore to reach the equivalence of $\hat{s}'$ and $s$ and $\hat{M}'$ and $M$ respectively.

Using appropriate starting values $c_1$ and $d_1$ one can now determine the $x_i'$ which corresponds to the empirically determined $\mu_i$. Hence

$$\hat{M}' = \frac{1}{n} \Sigma x_i \tag{24}$$

$$\hat{s}' = \frac{1}{n-1} \sum_{i=1}^{n-1} (x_{i+1}' - x_i') \tag{25}$$

$$= \frac{x_n - x_1}{n-1}$$

If the sums of the deviations do not exceed a certain $\varepsilon$, then the estimate is accepted as sufficiently exact :

$$|\hat{M}'' - \hat{M}'| \leq \varepsilon_M \quad (26)$$

$$|\hat{s}'' - \hat{s}'| \leq \varepsilon_s \quad (27)$$

If this is not the case the interval of the base variable is estimated which corresponds to the (0,1) interval of the membership values. To this end an upper bound $\bar{x}'$ and a lower bound $\underline{x}'$ is determined.

$$\bar{x}' = \hat{M}' + \frac{n}{2}\hat{s}' \quad (28)$$

$$\underline{x}' = \hat{M}' - \frac{n}{2}\hat{s}' \quad (29)$$

Now the corresponding $\bar{u}'$ and $\underline{u}'$, respectively, are computed and new parameters $\hat{c}$ and $\hat{d}$ are estimated. Experience has shown that it takes usually less than 10 iterations to reproduce the values of the base variable up to an accuracy of three units behind the decimal point. As starting points we used

$$c_1 = \frac{1}{n}\Sigma\mu_i \quad (30)$$

$$d_1 = \min(1-\frac{1}{k}, 2(1-c), 2c) \quad (31)$$

Where n is the number of stimuli and k is the number of different degrees of membership. If only the values 0 and 1 occur $d_1 = 1/2$.

Only the "linear" interval in the middle of the logistic function is used. With increasing k, d converges to 1 , i.e. $\lim_{k \to \infty} d =$ .
The entire range of the function is used. Finally it should be mentioned that not only monotonic functions, such as discussed so far, can be described but also unimodal functions by representing them by an increasing ($S_I$ ) and a decreasing ($S_D$) part.
Formally they can be represented as the minimum or maximum, respectively, of two monotonic membership functions each :

$$\mu_{S_I S_D}(x) = \min {}^{0}\left[\mu_{S_I}(x), \mu_{S_D}(x)\right]^{1} \qquad (32)$$

$$\mu_{S_I S_D}(x) = \max {}^{0}\left[\mu_{S_I}(x), \mu_{S_D}(x)\right]^{1} \qquad (33)$$

A computer program was written to process the observed data.

## Empirical Evidence

64 subjects (16 for each set) from 21 to 25 years of age individually rated 52 different statements of age concerning one of the four fuzzy sets "very young man" (vym), "young man" (ym), "old man" (om) and "very" old man" (vom).

The evaluation of the data showed a good fit of the model. Figures 8 -13 show the membership functions given by six different persons. As can be seen, the concepts "vym" and "ym" are realized in the monotonic type as well as in the unimodal.

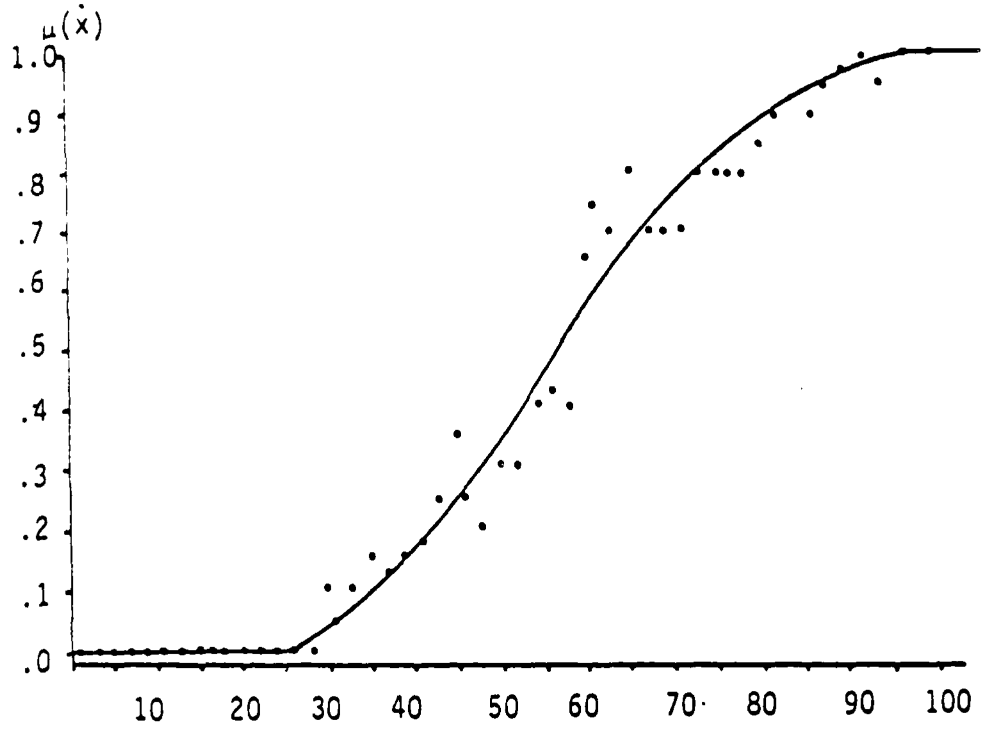



Fig. 8 : Subject 34, "old man"

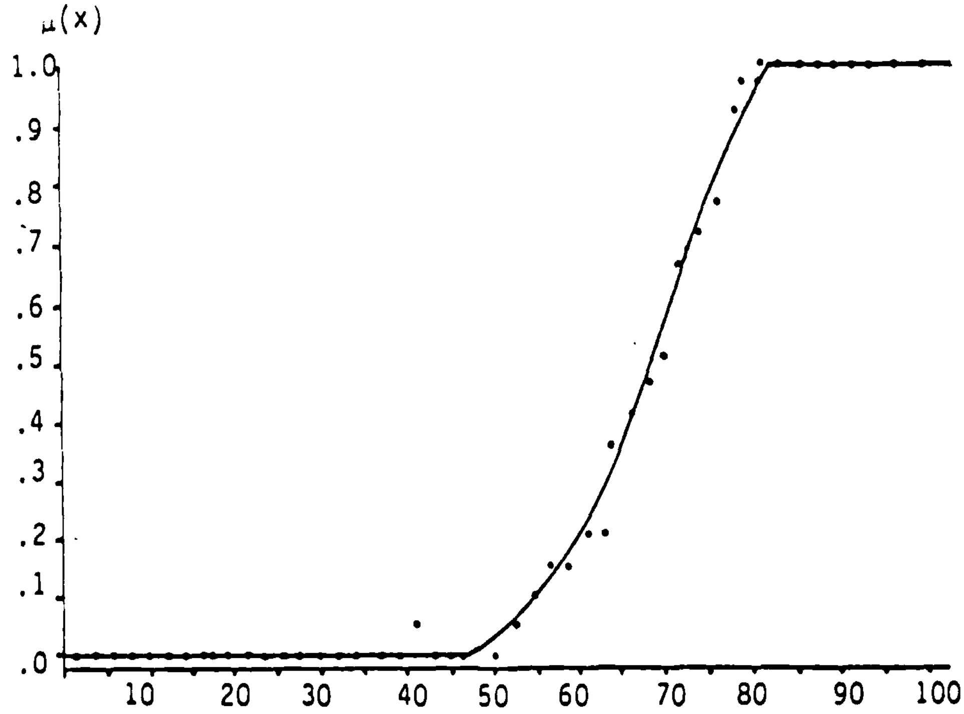



Fig. 9 ; Subject 58, "very old man"

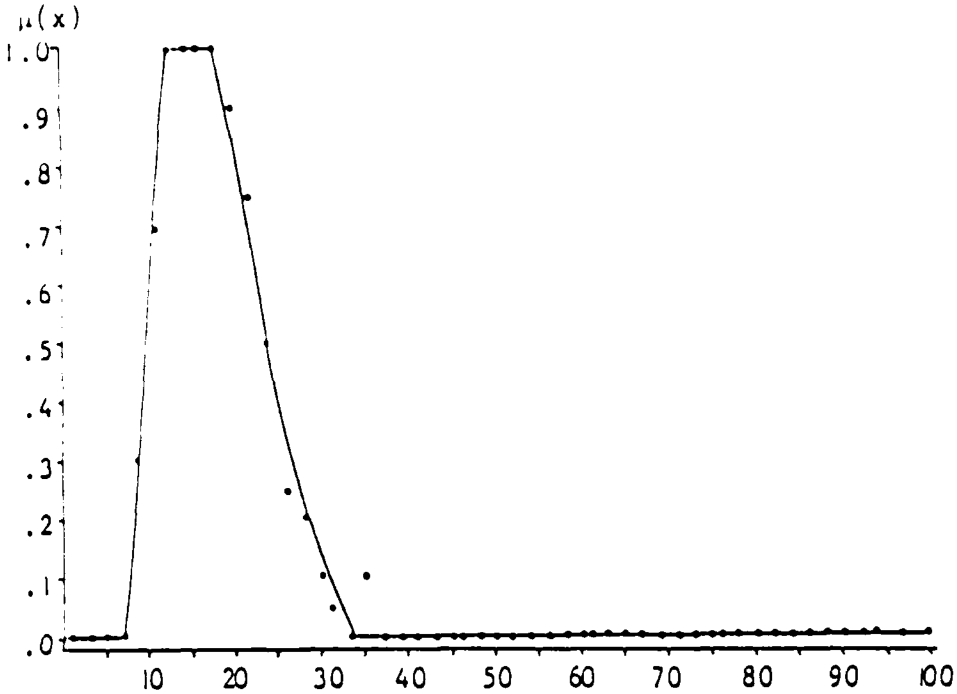



Fig. 10 :Subject 5, "very young man"

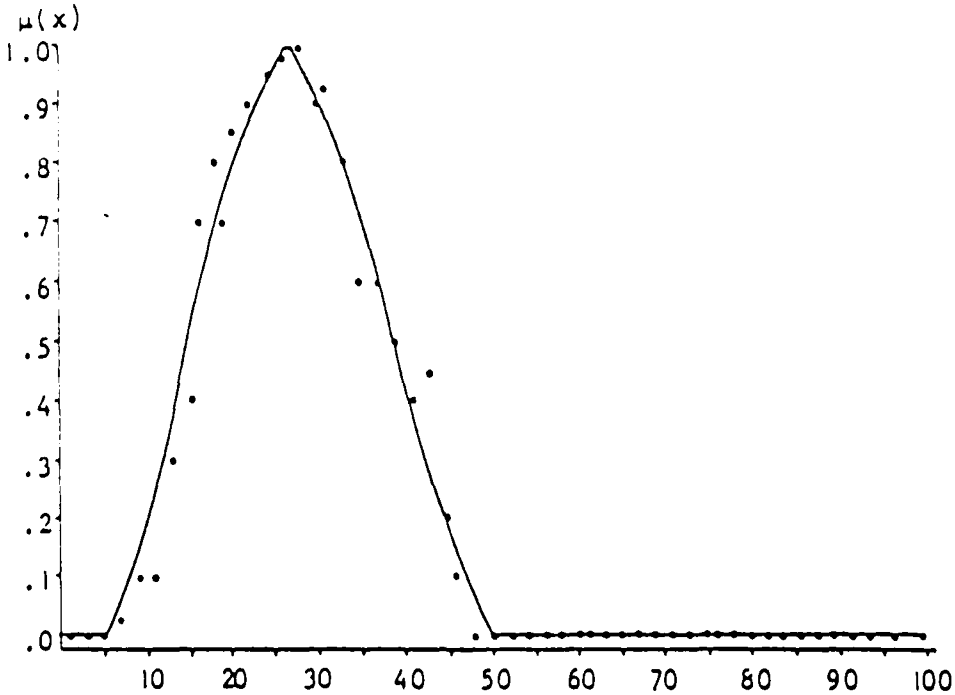



Fig. 11: Subject 17, "young man"

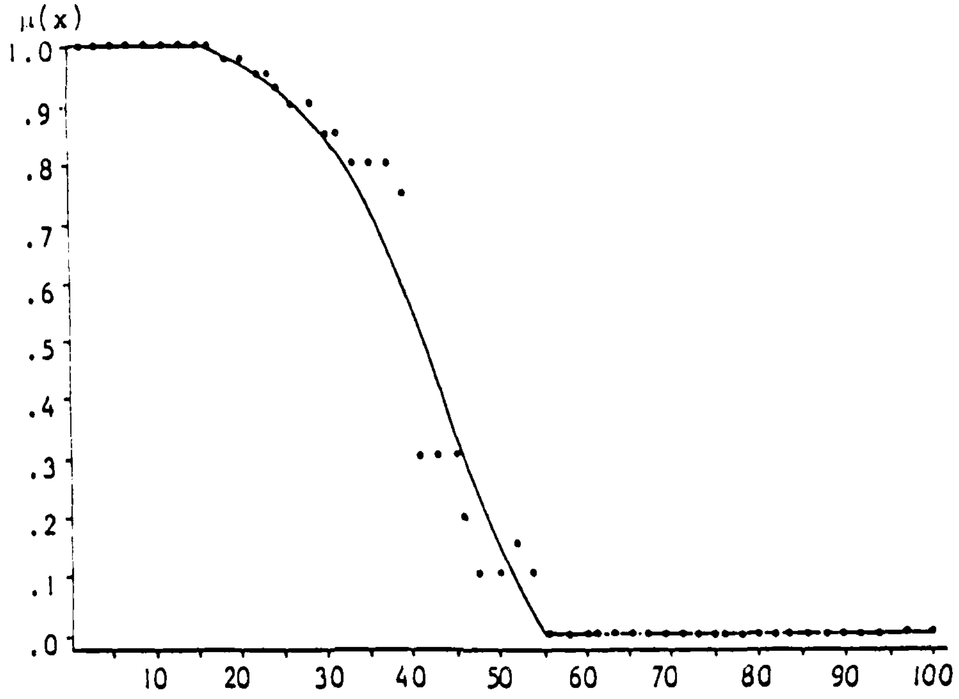



Fig. 12: Subject 15, "very young man"

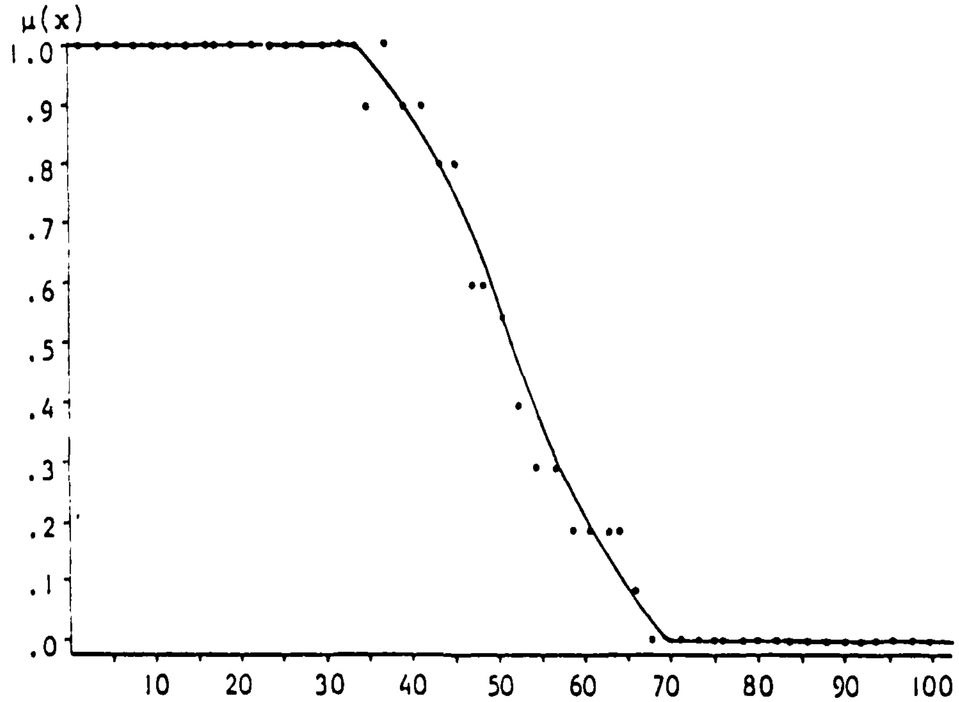



Fig. 13 : Subject 32, "young man"

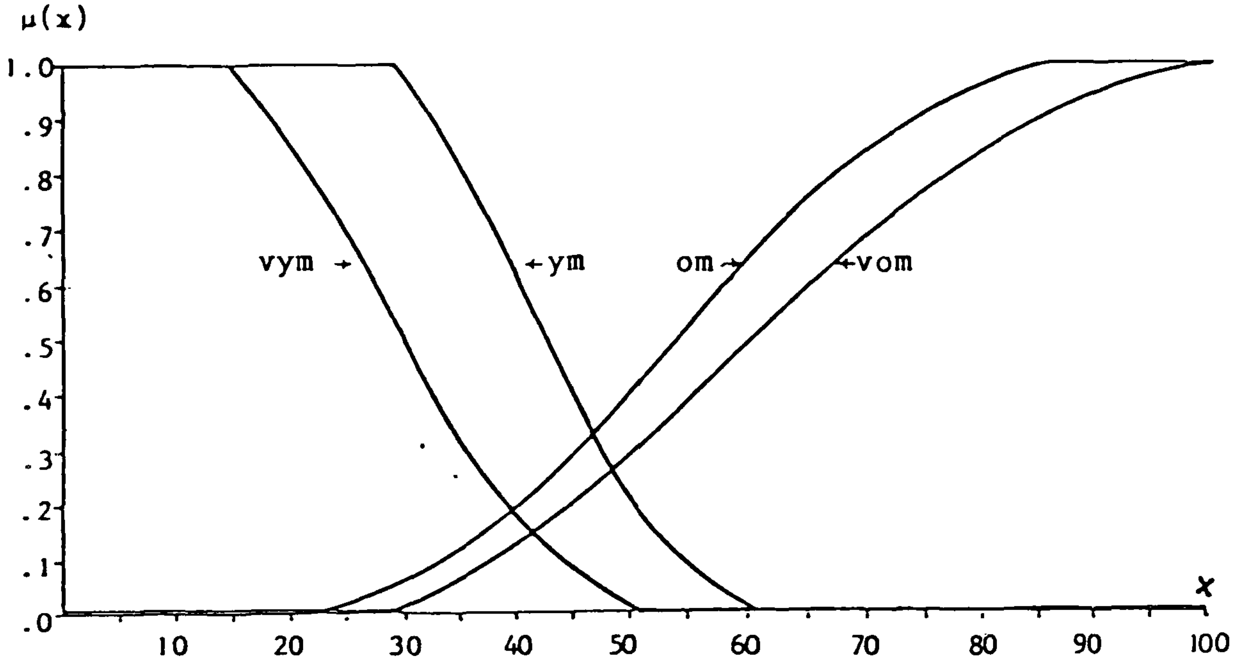



Fig. 14 : Generalized membership function (monotonic type) "very young man" (vym), "young man" (ym), "ôld man" (om) and "very old man" (vom)

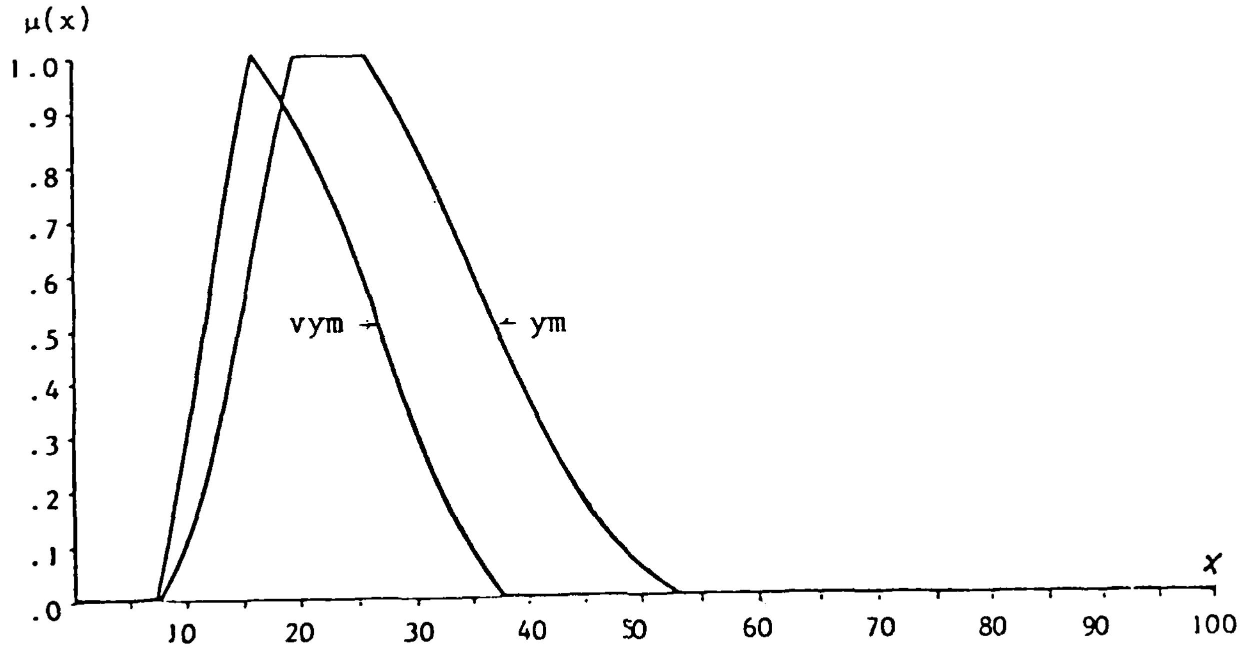



Fig. 15 : Generalized membership function (unimodal type) for "very young man" (vym) and "young man" (ym)

One may ask whether a general membership function for each of the four sets can be established. Even though the variety of conceptual comprehension is rather remarkable, there should be an overall membership function at least in order to have a standard of comparsion for the individuals. This is achieved by determining the common parameter values a, b, d and d for each set. Obiously the general membership functions of "old man" and "very old man" (Fig. 14) are rather similar. They practically differ only with respect to their inflection points, indicating a difference of about five years between "old man". The same holds for the monotonic type (Fig. 14) of "very young man" and "young man"; Their inflection points differ by nearly 15 years. It is interesting to note that the modifier "very" has a greater effect on "young" than on "old", but in both cases it can be formally represented by a constant. Several subjects provided the unimodel type in connection with "very young" and "young". Again the functions show a striking congruency (Fig. 15).

Of the slope is an indicator for vagueness (Kochen & Badre 1973) then the meaning of "young" is less vague than that of "old". On the other hand, the variability of membership functions may be regarded as an indicator of ambiguity. Thus, though being less vague, "young" seems to be more ambigous.

## III.2. Aggregation Operators

As already mentioned in section II.1 a decision in a "fuzzy enviremont" has been defined as the intersection of fuzzy sets representing either objectives or constraints. The grade of membership of an object in the intersection of two fuzzy sets, i.e. the fuzzy set "decision", was determined by use of either the min-operator or the product operator. The following example is an illustration of this :

Example 3 : The board of directors is trying to find the "optimal" dividend to be paid to the shareholders. For financial reasons it ought to be attractive and for reasons of wage negotiations it should be modest (Fig. 16).

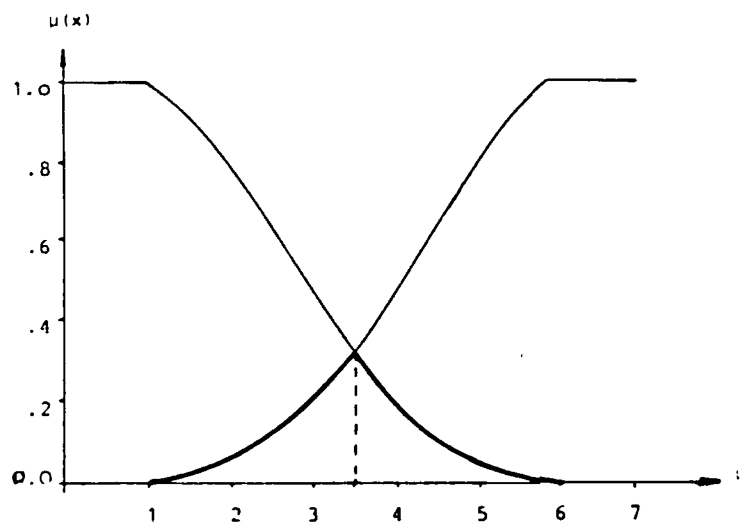



Figure 16 : fuzzy decision; x = dividend (%)

The optimal dividend to be paid to the shareholders would be 3.5%, considering the dividend with the highest degree of membership in the fuzzy set "decision" as the "most desirable".

Rather than viewing a decision as the intersection of several fuzzy sets (Thole, Zimmermann, Zysno 1979) one could describe it also as the union of all relevant fuzzy sets, using the maximum operator for aggregation :

Example 4 : An instructor at a university has to decide how to grade written test papers. Let us assume that the problem to be solved in the test was a linear programming problem and that the student was free to solve it either graphically or using the simplex method. The student has done both. The student's performance is expressed - for graphical solution as well as for the algebraic solution - as the achieved degree of membership in the fuzzy sets "good graphical solution" (G) and "good simplex solution" (S), respectively.
Let us assume that he reaches

$$\mu_G = 0.9 \quad \text{and} \quad \mu_S = 0.7.$$

If the grade to be awarded by the instructor corresponds to the degree of membership of the fuzzy set "good solutions of linear programming problems" it would be quite conceivable that this grade $\mu_{LP}$ could be determined by

$$\mu_{LP} = \mathrm{Max}(\mu_G, \mu_S) = \mathrm{Max}(0.9, 0.7) = 0.9$$

The two definitions of decisions - as the intersection or the union of fuzzy sets - imply essentially the following :

The interpretation of a decision as the intersection of fuzzy sets implies no positive compensation (trade-off) between the degrees of membership of the fuzzy sets in question, if either the minimum or the product is used as an operator. Each of them yields degrees of membership of the resulting fuzzy set (decision) which are on or below the lowest degree of membership of all intersecting fuzzy sets (see Example 3).

The interpretation of a decision as the union of fuzzy sets, using the maxoperator, leads to the maximum degree of membership achieved by any of the fuzzy sets representing objectives or constraints. This amounts to a full compensation of lower degrees of membership by the maximum degree of membership (see Example 4).

Observing managerial decisions one finds that there are hardly any decisions with no compensation between either different degrees of goal achievement or the degrees to which restrictions are limiting the scope of decisions. The compensation, however, rarely ever seems to be "complete" such as would be assumed using the max-operator. It may be argued that compensatory tendencies in human aggregation are responsible forthe failure of some classical operators (min, product, max) in empirical investigations (Hersh & Caramazza 1976; Thole, Zimmermann & Zysno 1979).

Neither the non-compensatory "and" represented by operators which map between zero and the minimum degree of membership nor the fully compensatory "or" represented by operators which map between the maximum degree of membership and 1 are appropriate to model the aggregation of fuzzy sets representing managerial decisions.

New additional operators will have to be defined which imply some degree of compensation, i.e. which map also between the minimum degree of membership and the maximum degree of membership of the aggregated sets. By contrast to modelling the non-compensatory "and" or the fully-compensatory "or" they should represent types of aggretation which we shall call "compensatory and".

It is possible that human beings use many non-verbal connectives in their thinking and reasoning. Being forced to verbalize them men possibly map the set of "merging connectives" into the set of the corresponding language connectives ("and", "or"). Hence, when talking, they use the verbal connective which they feel closest to their "real" non-verbal connective.

In analogy to the verbal connectives, the logicians defined the connectives "$\wedge$" and "$\vee$", assigning certain properties to each of them. By this, compound sentences can be examined for their truth values. In contrary to this constructive process, the empirical researcher has to analyze a given structure.

For the generation of promising and testable models we considered the relationships between different levels of the hierarchies mentioned in Section II 2.
Intensionally, in set theory higher level concepts are defined by the union of the attributes of lower level concepts. Extensionally, however, higher level concepts equal the intersection of corresponding lower level concepts (Zysno 1980). The most popular algebraic representation of this type of aggretation is the Minimum :

$$\mu_\theta = \mathrm{Min}(\mu_i) \tag{34}$$

where $\mu_i$ (x) is the grade of membership of element x to set $A_i$ (for convenience, x and A are dropped in the formulas); $x \in X$ = Universe of realized entities; $\theta$ is the fuzzy set representing an empirical supercategory : $A_1, A_2, \ldots, A_i \ldots, A_m \subset \theta \subset X$.

However, operators like this yield acceptable predictions only in very special situations. This probably is due to the tendency of man to compensate attribute deficiences of one aspect by stressing certain attributes of another aspect.

In extremal situations complete compensation is possible; in this case the maximum operator would seem appropriate.

$$\mu_\theta = \mathrm{Max}(\mu_i). \tag{35}$$

In order to model human evaluative behavior the pool of candidates to be tested should comprise such operators which work between minimum and maximum. Of course, they should also satisfy the desirable mathematical requirements of continuity, strict monotonicity , injectivity in each argument(which is implied by the presence of continuity and monotonicity), commutativity (which is implied by the presence of continuity, injectivity, and associativity (Aćzel 1961)).

Unfortunately, it is hard to find an averaging operator meeting all these requirements. Therefore,we should abandon at least one of them. Most critical seems to be the associativity as it is fulfilled by the median (Fung & Fu 1975) only. Hence we will be flexible with respect to this property.

Simple and well known operators regarding the remaining mathematical requirements are the geometric mean

$$\mu_0 = \left(\prod_{i=1}^{m} \mu_i\right)^{-m} \tag{36}$$

and arithmetic mean :

$$\mu_0 = \frac{1}{m} \sum_{i}^{m} \mu_i \tag{37}$$

An example aggregating two membership functions by each of the four operators is given in Figure 17.

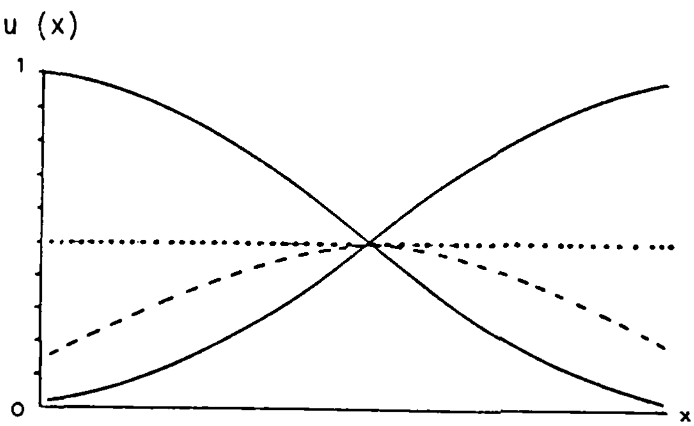



Fig. 17 : Aggregation of two membership functions by geometric (arithmetic)means as depicted by the dashed (dotted) line.

Experiments (Zimmermann & Zysno 1979, 1980, 1983) conducted in order to get empirical evidence on this problem lead to the following conclusions :

(1) People use averaging operations when making judgments or evaluations resulting in membership values between minimum and maximum.
(2) The geometric mean and to some extent the arithmetic mean are adequate models for human aggregation of fuzzy sets when special compensatory effects exist.
(3) Men use still other connectives than "and" and "or".

Quite naturally, if several operators are necessary in order to describe a variety of phenomena, the question crises, how many operators are needed, as each important situation in practice would then call for an adequate model. Moreover, one would be forced to assume that man has a decision rule enabling him to choose the right connective for each situation. The pursuit of this train of thought and especially its application implies a lot of difficulties.
We feel that one way to bypass these difficulties is to generalize the classical concept of connectives by introducing a parameter which may be interpreted as "grade of compensation".
Each point on the continuum between "and" and "or" represents a different operator.
One way to formalize this idea is to find an algebraic representation for a weighted combination of the non-compensatory "and" and the fully compensatory "or" : The more there is a tendency for compensation the more the "or" becomes effective and vice versa.

Let X be the universe of discourse with the elements x. A, B and $\Gamma$ are fuzzy sets in X. Then, the convex combination of A, B and $\Gamma$ can be denoted by $(A, B; \Gamma)$ and is defined either by the relation

$$(A, B; \Gamma) = \bar{\Gamma}A + \Gamma B \tag{38a}$$

or

$$(A, B; \Gamma) = A^{\bar{\Gamma}} \cdot B^{\Gamma} \tag{38b}$$

where $\bar{\Gamma}$ is the complement of $\Gamma$. Written in terms of membership functions, (38) reads

$$f_{(A,B;\Gamma)} = (1-f_{\Gamma}(x)) \cdot f_A(x) + f_{\Gamma}(x) \cdot f_B(x) \tag{39a}$$

and

$$f_{(A,B;\Gamma)} = f_A(x)^{1-f_{\Gamma}(x)} \; f_B(x)^{f_{\Gamma}(x)} \tag{39b}$$

A basic property of the above-defined convex combination is expressed by :

$$A \cap B \subset (A,B;\Gamma) \subset A \cup B$$

Obviously, the convex combination is a fuzzy set between the intersection and the union of two fuzzy sets A and B.

one model, fulfilling the above mentioned properties and having performed as the best balanced representation so far in several experiments including practical decision situations is the so called $\gamma$- operator

$$\mu_{\Theta} = \left(\prod_{i=1}^{m} \mu_i^{\delta_i}\right)^{1-\gamma}\left(1-\prod_{i=1}^{m}(1-\mu_i)^{\delta_i}\right)^{\gamma} \qquad (40)$$

This model is a convex combination of the product and the algebraic sum, which are known as algebraic representations of the intersection and the union, respectively.

The $\gamma$- operator seems rather complicated especially for use in linear models. Thus additional compensatory operators have been considered. The convex combination of minimum and maximum operator

$$\mu_{\theta} = (1-\gamma) \min_{i=1}^{m} \{\mu_i\} + \gamma \max_{i=1}^{m} \{\mu_i\} \quad \gamma \in [0,1] \qquad (41)$$

is a special case of relation in which the min-operator stands for "and" the max-operator for "or". $\gamma$ again is the parameter of compensation. Using the empirical data of Zimmermann, Zysno (1983) this operator gives a rather good model for aggregation although the $\gamma$-model is better.One of its advantages is its computational simplicity. A slight disadvantage could be seen in the fact that extreme values get a higher weight and that dominated solutions cannot be recognized after aggregation.

Example :

Consider three alternatives $x_1$, $x_2$, $x_3$ each with membership values of

$\mu_1$, $\mu_2$ and $\mu_3$ :

| | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| $\mu_1$ | 0.7 | 0.8 | 0.8 |
| $\mu_2$ | 0.3 | 0.7 | 0.4 |
| $\mu_3$ | 0.1 | 0.4 | 0.4 |

| | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| min | 0.1 | 0.4 | 0.4 |
| max | 0.7 | 0.8 | 0.8 |

Although $x_3$ is strictly dominated by $x_2$ the aggregation $\gamma \min + (1-\gamma) \max$ gives the same result for both and so both are elements of the optimal decision. Additional considerations are necessary in such a case to exclude those results. This fact is the consequence of minimum and maximum being not strictly monotonous and so the convex combination is not either.
The following figure shows the compensatory effect of this operator for different values of $\gamma$.

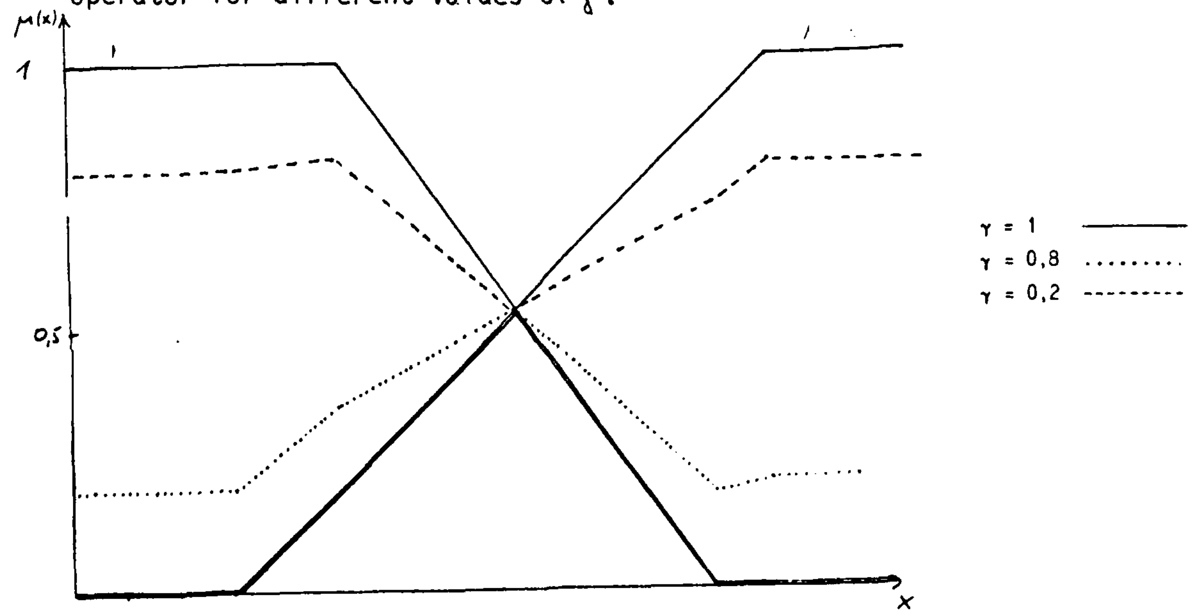



Fig. 18
Aggregation of two membership functions by a convex combination of min - max

To avoid this effect and to give higher weight to middle values new operators are proposed by Werners (1984). The idea is to differentiate between the terms "and" and "or", to allow compensation and to get the minimum when expressing the logical "and" and maximum when intending the logical "or" :

$$\mu_{\underset{\sim}{and}} = \gamma \min_{i=1}^{m} \mu_i + (1-\gamma)\frac{1}{m}\sum_{i=1}^{m}\mu_i \qquad \gamma \in [0.1] \tag{42}$$

$$\mu_{\underset{\sim}{or}} = \gamma \max_{i=1}^{m} \mu_i + (1-\gamma)\frac{1}{m}\sum_{i=1}^{m}\mu_i \qquad \gamma \in [0.1] \tag{43}$$

Here $\gamma$ is the degree of approximating the logical meaning of "and" and "or", respectively. The arithmetic mean gives a compensatory effect. $\gamma = 1$ yields $\mu_{\underset{\sim}{and}} = \min$ and $\mu_{\underset{\sim}{or}} = \max$. The combination of these two operators leads to very good results with respect to the empirical data of Zimmermann, Zysno (1983). The mathematical structure seems to be rather easy and efficiently to be handled. Both operators are communative, idempotent, monotonous, continuous, compensatory and generalisations of the logic "and" and "or", respectively (Werners 1984). The following figures illustrate these operators :

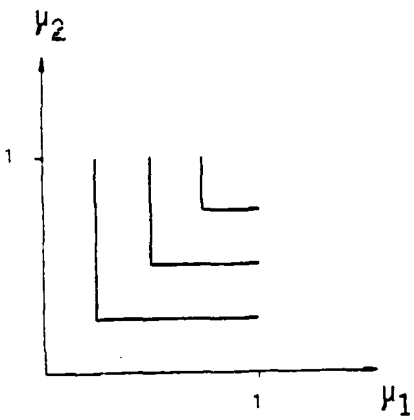



Fig. 19: and, $\gamma = 1$

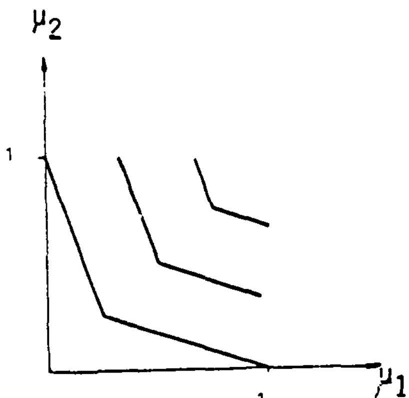



Fig. 20 : and, $\gamma = \frac{1}{2}$

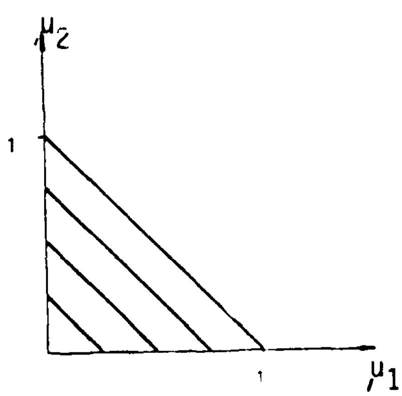



Fig. 21 : and, $\gamma = 0$

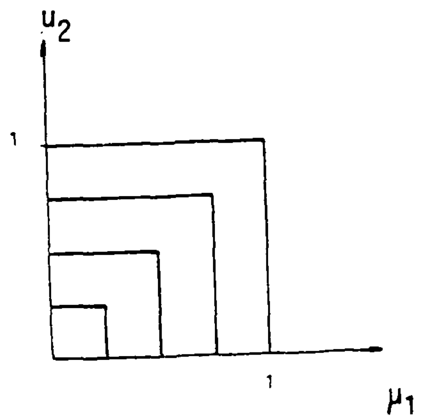



Fig. 22 : or, $\gamma = 1$

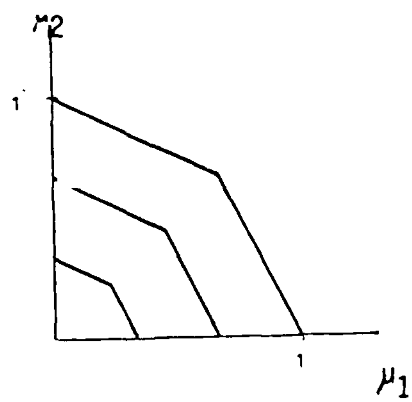



Fig. 23 : or, $\gamma = \frac{1}{2}$

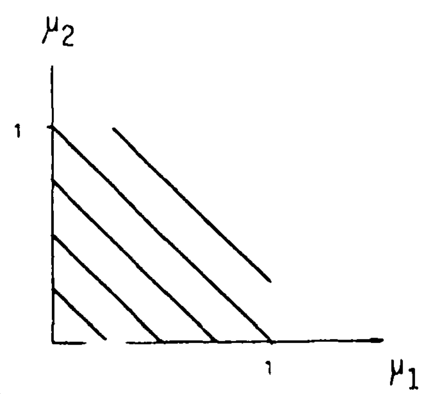



Fig. 24 : or, $\gamma = 0$

## III.3 Mathematical Programming Model

### III.3.1 Combination : Operator and Membership Function

Considering different types of membership functions, possible aggregation operators and feasable algorithms 10 model types can be characterized in terms of the results. The numbers of the columns refer to the "hierarchy of decisions" as shown on page 15

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9a | 9b |
|---|---|---|---|---|---|---|---|---|---|---|
| Solution: Skalar | x | x | | | | x | x | x | x | x |
| Vektor | | | | | | x | x | x | x | x |
| Sorting | | | | x | x | | | | | |
| μ-Function | | | x | | | | | | | |
| Necessary M-F. | | | | | | | | | | |
| I | x | | | x | | x | x | x | | |
| II | | x | x | | x | | x | | Lin x | N-Lin x |
| Possible Aggregator | | | | | | | | | | |
| γ | x | x | x | | | | | | | |
| Min/Max | x | x | x | | | | | | x | |
| Min | x | x | x | | | | | | | x |
| others | x | x | x | | | | | | | |
| Algorithm | Aggr | Aggr | Aggr | sorting | FM sort | ? | ? | ? | MILP | LP |

This project focusses on mathematical programming models which can be solved efficiently. The computational effort depends essentially on the mathematical character of the "equivalent crisp model"

(last row in above table). The basic type of fuzzy model gives only one "degree of freedom".
The type of equivalent model depends much more on the type of membership function assumed and the type of operators used.

The "derived" models ought to be either linear programs or mixed integer linear programs, otherwise computions can become excessively long.

APEX, for instance, is an efficient tool for solution. Hence, our attention is directed towards models of type 9a and 9b.

Principally several combinations of membership functions and operators as proposed in III.1 and III.2 are possible. But the following discussion will show that only few of them can be solved efficiently.

The $\gamma$-operator, though empirically the most satisfying connective, leads to crisp equivalents which are extremely hard to solve. They are convex in some ranges, concave in other ranges and there do not seem to be efficient numerical algorithms available which could be used in the framework of an interactive decision support system to find optimal solutions efficiently. The following picture gives an impression of the unpleasant structure of these types of problems.

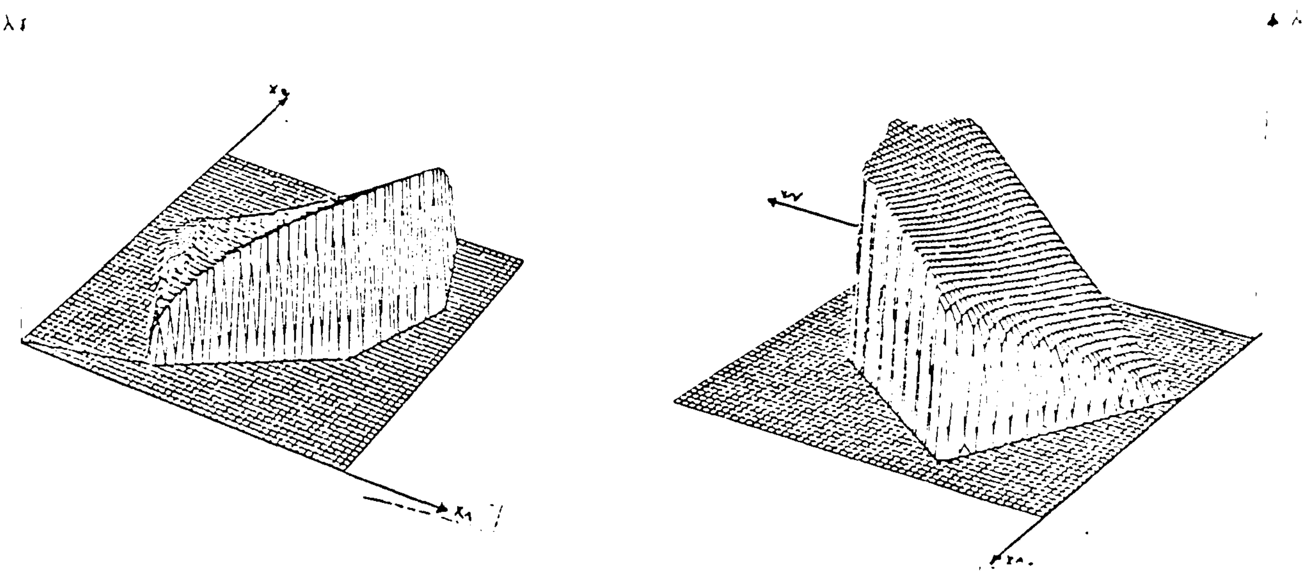


Fig. 25

The spline function as membership function together with the algebraic mean as a connective leads to crisp equivalent which are also non-linear but which could be solved by using gradient methods. These, however, do not seem to be well-suited for interactive approaches.
The improved logistic function with four parameters is too complicated to be used in mathematical programming models, especially no equivalent linear model can be found in general.
The logistic function, however, can be transformed such that a linear function results (using logarithms). Therefore we concentrate on linear original or transformed membership functions.

The feasability of computations does not only depend on the type of membership function but rather on the chosen combination of membership function and operator. With respect to operators we found that all those basing on product-type models lead to very nasty equivalent models.
So we concentrate on the operators minimum and the convex maximum and the new operators and and or.

Let us consider linear functions $c_i{}^t x$, on which membership functions $u_i$ are defined, i = 1,...m. Using the minimum operator an optimal decision x can be determined by solving

$$\max_{x \in X} \min_{i=1}^{m} \mu_i (c_i{}^t x) \tag{44}$$

This is equivalent to the mathematical programming model

$$\begin{aligned} &\max \alpha \\ &\quad \alpha \leq \mu_i (c_i{}^t x) \quad i = 1,\ldots,m \\ &\quad x \in X \end{aligned} \tag{45}$$

If $\mu_i$ is linear for all i , then (45) is a linear programming model and can be solved by an efficient linear programming code.
If the membership functions are nonlinear under certain conditions equivalent linear models can be derived. For the logistic membership function

$$\mu_{Li} = \frac{1}{1+e^{-a(x-b)}}$$

(45) becomes

$$\begin{aligned} &\max \alpha \\ &\text{s.th.} \alpha \leq \frac{1}{1+e^{-a_i(c_i{}^t x - b_i)}} \qquad \forall\, i=1,\ldots,m \\ &\quad x \in X \end{aligned} \tag{46}$$

If $(\alpha^0, x^0)$ is an optimal solution of (46) then $\alpha^0$ is the degree of membership of $x^0$ to the fuzzy set decision. But in this form the model

can hardly be solved. An equivalent formulation with

$$\alpha' = \ln\left(\frac{\alpha}{1-\alpha}\right)$$

is given by

$$\begin{aligned} &\max \alpha' \\ &\text{s.th. } \alpha' \le a_i(c_i{}^t x - b_i) && \forall i = 1,\dots,m \\ &\qquad x \in X \end{aligned}$$

$$\Leftrightarrow \quad \begin{aligned} &\max \alpha' \\ &\text{s.th. } \frac{1}{a_i}\alpha' - c_i{}^t x \le - b_i && \forall i = 1,\dots,m \\ &\qquad x \in X \end{aligned} \tag{47}$$

After solving (47) which is a normal LP the optimal solution of this model, $(\alpha^{o'}, x^o)$ has to be transformed to find the optimal solution to (46) by $\left(\frac{e^{-\alpha'o}}{1+e^{-\alpha'o}}, x^o\right)$.

Because the minimum operator does not allow any compensation a number of compensatory operators are considered in the following :

Using the convex combination of minimum and maximum the problem of finding an optimal alternative reads :

$$\max_{x \in X} \left( \gamma \min_{i=1}^{m} \left[\mu_i(x)\right] + (1-\gamma) \max_{i=1}^{m} \left[\mu_i(x)\right] \right) \tag{48}$$

$\gamma \in [0.1]$ degree of compensation

respectively for linear goals and constraints :

$$\max_{x \in X} \left( \gamma \min_{i=1}^{m} \{\mu_i(c_i{}^t x)\} + (1-\gamma) \max_{i=1}^{m} \{\mu_i(c_i{}^t x)\} \right) \tag{49}$$

Equivalent to (49) is

$$\begin{aligned} & \max \gamma\alpha_1 + (1-\gamma)\alpha_2 && && (50) \\ & \text{s.th. } \alpha_1 \le \mu_i(c_i{}^t x) && \forall i=1,\dots,m \\ & \qquad \alpha_2 \le \mu_i(c_i{}^t x) && \text{for at least one } i \in \{1,\dots,m\} \\ & \qquad x \in X \end{aligned}$$

In this mathematical programming model the second group of constraints can be substituted by

$$\begin{aligned} & \alpha_2 \le \mu_i(c_i{}^t x) + M\, y_i && \forall i=1,\dots,m \\ & \sum_{i=1}^{m} y_i \le m-1 && && (51) \end{aligned}$$

$y_i$ binary variable,

$M$ very large (dependent on the computer used)

Thus (50) can be modelled by a mixed-integer programming model. If all membership functions $\mu_i$ are linear in $c_i{}^t x$ then the resulting model is a mixed integer linear programming model (MILP) which can efficiently be solved by standard software, for instance APEX.
The same does not necessarily hold for non-linear membership functions, as can be shown for logistic functions. Then an equivalent model to (50)

is :

$$\max \; \gamma \frac{1}{1+e^{-\alpha_1'}} + (1-\gamma) \frac{1}{1+e^{-\alpha_2'}} \tag{52}$$

s.th. $\alpha_1' \leq a_i(c_i{}^t x - b_i)$ $\forall i=1,\ldots,m$

$\alpha_2' \leq a_i(c_i{}^t x - b_i)$ for at least one $i = 1,\ldots,m$.

$x \in X$

which is non-linear.

Considering the new operator "fuzzy and" with

$$\max_{x \in X} \left( \gamma \min_{i=1}^{m} \mu_i(c_i{}^t x) + (1-\gamma) \frac{1}{m} \sum_{i=1}^{m} \mu_i(c_i{}^t x) \right) \tag{53}$$

an equivalent programming model can be formulated :

$$\max \quad \alpha + (1-\gamma) \frac{1}{m} \sum_{i=1}^{m} \alpha_i \tag{54}$$

s.th. $\alpha + \alpha_i \leq \mu_i(c_i{}^t x)$ $\forall i=1,\ldots,m$

$x \in X$

$\alpha + \alpha_i \leq 1$ $\forall i=1,\ldots,m$

$\alpha, \alpha_i \geq 0$

If $\mu_i(c_i{}^t x)$ are linear for i=1,...,m then (54) is a crisp LP.

With $\mu_i(x) = \mu_{L_i}(x) = \dfrac{1}{1 + e^{-a_i(c_i{}^t x - b_i)}}$ : (55)

$$\max \quad \alpha + (1-\gamma)\frac{1}{m}\sum_{i=1}^{m} \alpha_i$$

$$\text{s.th.} \quad \alpha + \alpha_i \leq \frac{1}{1 + e^{-a_i(c_i{}^t x - b_i)}}$$

$$x \in X$$

$$\alpha + \alpha_i \leq 1$$

$$\alpha, \alpha_i \geq 0$$

A substitution leads to the following crisp mathematical programming model with linear constraints and nonlinear objective function.

$$\max \quad \frac{1}{1 + e^{\alpha'}} + (1 + \gamma)\frac{1}{m}\sum_{i=1}^{m} \frac{e^{\alpha'}(1 - e^{\alpha_i'})}{1 + e^{\alpha'}(1 + e^{\alpha'} + e^{\alpha' + \alpha_i'})}$$

$$\text{s. th.} \quad \alpha' + \alpha_i' \leq a_i(c_i{}^t x - b_i) \qquad \forall_i = 1,\dots,m$$

$$x e^{X}$$

$$\alpha', \alpha_i' \geq 0$$

(56)

$\alpha + \alpha_1 \leq 1$ in (55) $\forall_i = 1,\dots,m$ follows from

$$\alpha + \alpha_i = \frac{1}{1 + e^{\alpha' + \alpha_i'}} \leq 1 \quad \text{for } e^{\alpha' + \alpha_i'} \geq 0 \, \forall \alpha', \alpha_i'$$

The following table presents those combinations of membership functions and aggregation operators which lead to efficiently solvable models and can be introduced into a decision support system :

| | Min | Min/Max | A.M. | And | $\gamma$-Oper. |
|---|---|---|---|---|---|
| Linear Model | L.P. | Mix.Int. | L.P. | L.P. | |
| Logistic Model | L.P. | | | | |
| Ext.Log. Model | | | | | |

## III.3.2 Interactive Model

Using the decision support system first the decision maker has to give his goals and constraints for a fuzzy programming model. Goals and constraints are not treated equally as in the fuzzy decision model of Zadeh mentioned earlier. Instead we consider as the main difference between a fuzzy goal and a fuzzy constraint that the decision maker is able to give more information about a constraint than about a goal. Similar to crisp programming models where he only distinguishes between 0 and 1 degree of membership for satisfying a constraint the decision maker a priori gives a membership function for each constraint. The membership function of a fuzzy maximization goal cannot be given in advance but depends on what is possible when satisfying the constraints. So additional information has to be attained about the dependencies of the model. This can be done by the system. Here extreme solutions are determined optimizing one goal over two crisps feasable regions : one with degree of membership of one, the other with positive degree of membership until zero. The results are used to determin membership functions of the goals.
Solving a crisp vector maximum model Zimmermann (1978) proposed to deduce membership functions dependent on the ideal and the pessimistic solutions. The concept used here to propose membership functions for the goals is a generalization which is necessary to handle fuzzy goals under crisp and fuzzy constraints. Aggregating all membership values i.e. of goals and constraints a compromise solution is determined. Interactively the decision maker can now change the proposed membership functions until he is satisfied with the compromise solution.

The interactive fuzzy programming system supports a decision maker, especially in two different ways :

- first, it determines extreme solutions and proposes membership functions describing the goals.
- second, it evaluates efficient compromise solutions with additional local informations.

After each presented compromise the decision-maker gets more and more insight into the model and can articulate further preference information :

- local, by modifying membership functions,
- global, by modifying the model.

The following rough flow chart sketches how the DSS works :

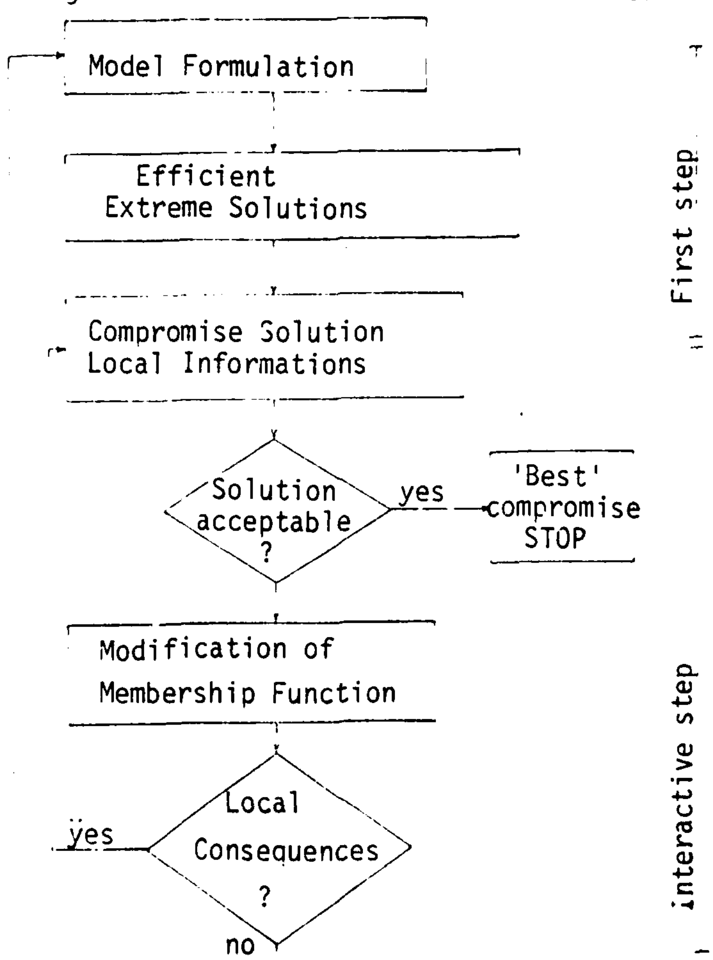



Figure 26 :
Rough flow chart DSS

The consequences of an interactive variation of a linear membership function can be seen in figs. 27a, 27b. Here $\mu_1$ is fixed whereas $\mu_2$ is modified by changing $\underline{c}$ to $\underline{c}'$.
The resulting compromise solution $x^{o'}$ after modification has a higher value $c^t x^{o'} > c^t x^o$, but the degree of membership has decreased. This is the result of the higher aspiration level formulated by the decision maker.

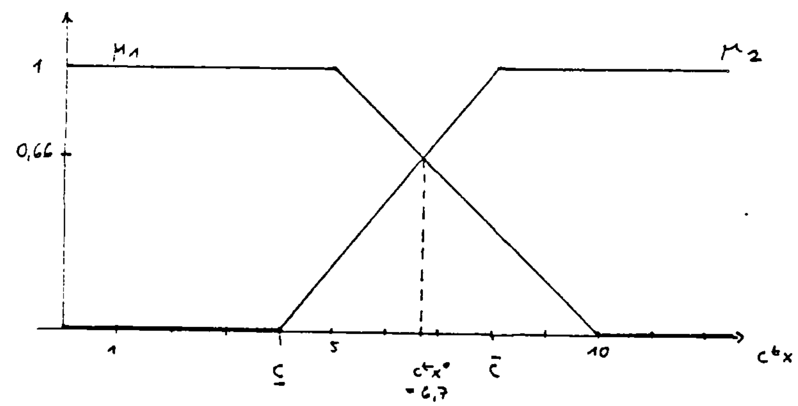



Fig. 27a

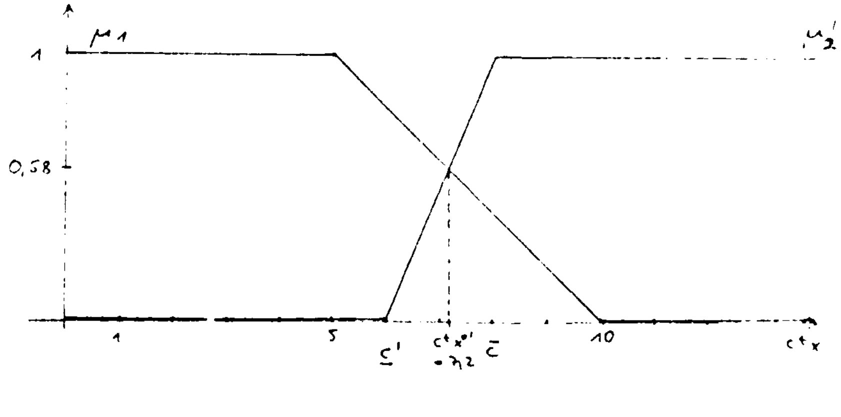



Fig. 27b

| | $c_1$ | $c_2$ | ... | $c_k$ |
|---|---|---|---|---|
| $x^{o1}$ | $\sum_j c_{1j}x_j^{o1}$ | $\sum_j c_{2j}x_j^{o1}$ | ... | $\sum_j c_{kj}x_j^{o1}$ |
| $\vdots$ | | | | |
| $x^{ok}$ | $\sum_j c_{1j}x_j^{ok}$ | $\sum_j c_{2j}x_j^{ok}$ | ... | $\sum_j c_{kj}x_j^{ok}$ |
| $x^{11}$ | $\sum_j c_{1j}x_j^{11}$ | $\sum_j c_{2j}x_j^{11}$ | ... | $\sum_j c_{kj}x_j^{11}$ |
| $\vdots$ | | | | |
| $x^{1k}$ | $\sum_j c_{1j}x_j^{1k}$ | $\sum_j c_{2j}x_j^{1k}$ | ... | $\sum_j c_{kj}x_j^{1k}$ |

Fig. 28 : Extreme solutions

$x^{o1},\ldots,x^{ok}$ and $x^{11},\ldots, x^{1k}$ are extreme solutions of a fuzzy maximization model with membership value to the constraints of 0 or 1 respectively.

Afterwards a compromise solution $x^0$ is determined by the system and is proposed to the decision maker including the degrees of membership to the different goals and constraints

Using the DSS the decision maker has to give his data for the following fuzzy linear programming model :

$$
\begin{array}{llll}
k_1 & \underset{\sim}{\max} & C_1\, x & \\
k_2 & \underset{\sim}{\min} & C_2\, x & \\
m_1 & \text{s.th.} & A_1\, x \underset{\sim}{\leq} \underline{b}_1,\ \bar{b}_1 & \\
m_2 & & A_2\, x \underset{\sim}{\geq} \underline{b}_2,\ \bar{b}_2 & \qquad\qquad (57) \\
m_3 & & A_3\, x \underset{\sim}{=} \underline{b}_3,\ b_3,\ \bar{b}_3 & \\
 & & D_1\, x \leq e_1 & \\
 & & D_2\, x \geq e_2 & \qquad x \in X \\
 & & D_3\, x = e_3 & \\
 & & \quad x \geq 0 &
\end{array}
$$

Assumption : $\underline{b}_i < b_i < \bar{b}_i \quad \forall i$

For each goal an efficient individual optimum is determined considering the constraints satisfied with membership degree one or zero respectively. The extrem solutions are presented to the decision maker in a table and are used to determine the membership functions of the different goals (theoretical evaluation to this point can be found in (Werners, 1984)) .

| | | | | | | |
|---|---|---|---|---|---|---|
| | | $\underset{\sim}{\max}$ | $\underset{\sim}{\min}$ | $\underset{\sim}{\leq}$ | $\underset{\sim}{\geq}$ | $\underset{\sim}{=}$ |
| | $\bar{c}_1$ | $\bar{c}_2$ | $\bar{c}_k$ | $\bar{b}_1^1$ ... | $\underline{b}_1^2$ ... | $b_1^3$ .... |
| $x^o$ | $c_1{}^t x^o$ | $c_2{}^t x^o$ | $c_k{}^t x^o$ | $a_1{}^{1t} x^o$ | $a_1{}^{2t} x^o$ | $a_1{}^{3t} x^o$ |
| | $\underline{c}_1$ | $\underline{c}_2$ | $\underline{c}_k$ | $\bar{b}_1^2$ ... | $\underline{b}_1^2$ | $\underline{b}_1^3$, $\bar{b}_1^3$ |
| $z^o$ | $\alpha^o+\alpha^o_{o1}$ | $\alpha+\alpha^o_{o2}$ | $\alpha+\alpha^o_{oh}$ | $\alpha+\alpha^o_{11}$ | $\alpha+\alpha^o_{21}$ | $\alpha+\alpha^o_{31}$ |

Fig. 29 : Compromise solution

Now the decision maker can decide whether he agrees with one of the proposed solutions or whether he wants to change one of the membership functions or the degree of compensation of the aggregation operator.

## III.4 Empirical Investigation : Portfolio Analysis

The DSS should be tested in real military or civilian decision situations. Besides the methodological restrictions mentioned in the previous chapter restrictions concerning reality and computation time had to be taken into account.
Thus the empirical decision situation had to be chosen such that

1. the substantial concepts have to be understood as fuzzy subjective categories and as fuzzy goals/restrictions,
2. the membership function is linear or logistic in first approximation,
3. the aggregation of the subjective categories and the fuzzy goals/restrictions respectively can be represented satisfactorilly by simple operators such as minimum and maximum convex combinations of both, or convex combinations of min and alg. mean.
4. the decision situation can be described by few categories so that it remains comprehensable,
5. the hierarchie of criteria (for the use in the LP) has the same depth in all branches,
6. the data can be collected with acceptable effort.

No military problem situation could be made available to us. We, therefore, turned the above mentioned non-military cases. Equirements seemed to be fulfilled in the decision situation when buying bonds. After having collected information concerning the quality of the alternatives from brokers or other sources the

decision maker generally selects relevant aspects like maximizing the profit or minimizing the risk. His goals and/or restrictions can be formulated in a fuzzy way formalized by membership functions.

Often the investor has to rely on brokers because of the complexity of the problem. Anyway the goals of the investor, his individual economic situation and the restrictions resulting from individual preferences should be clear.

Most of the investors want to maximize the annual profit no matter whether it results from raising stock prices or from dividends. Those who use the profit for subsistence prefer safe monetary returns and raise the portion of shares with high dividends or of bonds with fixed interests. A third group prefers an increase of the value of the portfolio and therefore tends to shares with growing values.

The economic situation of the investor restricts the budget. It also determines the portfolio-condition which is formalized by a single investment. For example the budget can be DM 100.000.- and the maximum for a single investment 10 per cent or DM 10.000.-

Restrictions concerning the individual preferences are normally stated in terms such as "more defensive/more aggresive" or "risk avoiding/speculative". They should be explicitly formulated, for example, as lower bounds for the increase in price and dividend,

"tolerable" fall in price or dividend. This yields substantial restrictions for the alternatives. So the investor learns about how realistic his expectations are and can correct them if there is no bond satisfying all his wishes.

### III.4.1 Models

When modelling the goals and restrictions of the investor on the one hand and the evaluation of the brokers on the other hand one needs a common formulation which can serve as a basis for the interface.

#### III.4.1.1 The descriptive model

First we have to create a simple system which is acceptable both for the investor and the broker. Within this system the process of valuation should be clearly structured so that the investor can at least partly understand the propositions of the broker in order to correct his goals if necessary. So the system should satisfy the following conditions :

1. Simplicity : An investor with normal education should be able to understand the system.
2. Substance : The system should contain the substantial aspects of evaluation.
3. Symmetry : The criteria of evaluation should be chosen such that both the structure of the expectations of the investor and the aspects of evaluation by the broker are represented.

The main goal of the evaluation is to reach a "good investment". This consists of an increase in prices and of an attractive dividend. For each of there aspects one can expect a more or less satisfactory development (supposing fixed environmental factors) which can be described by the price after an agreed period for one year. The development itself and so its forecast is uncertain which yields fluctuations in the stock-exchange prices. The analogous holds for the dividend which is not guaranteed to be stable.

The crucial problem of investment is the risk. The normal goal of the investor is a profit as high and as safe as possible. But there is hardly any bond with these qualities. The owners of such papers would have no interest to sell causing the prices to raise because of the great demand.

Papers with an uncertain development often have better opportunities of good profit combined however with a rather high risk. Now the above mentioned preference (concerning the risk) of the investor becomes decisive. Often a mixture of bonds with different opportunities of profit is recommanded according to the principle of diversification.

The analogous holds for the dividend which rarely is guaranteed to be stable. In the worst case, however, it can fall to 0 per cent which amounts to a loss when taking into consideration inflation.

The above described criteria can be ordered into a simple hierarchical structure as to their dependencies (Fig. 30 ). The rate of interest is supposed to be stable in order to facilitate the scheme.

(The corresponding lines in the hierarchy are therefore dashed).

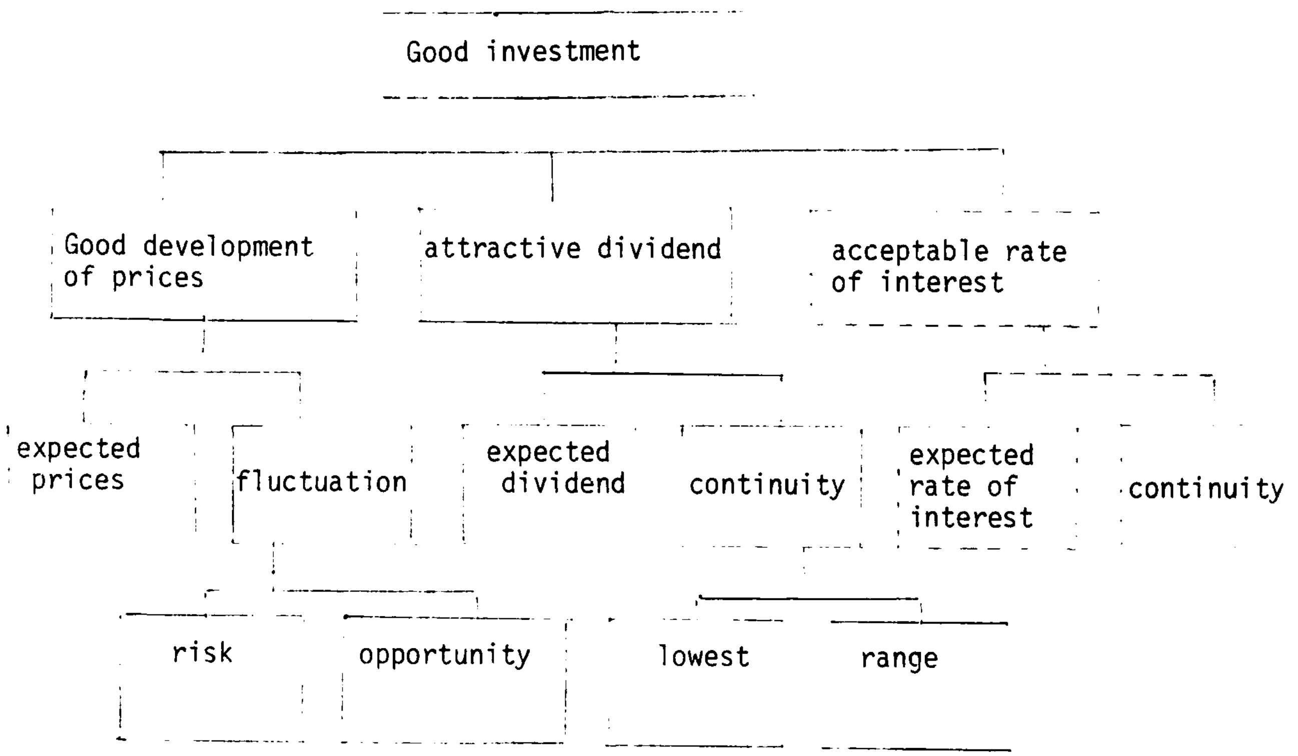



Fig. 30 : Hierarchy of criteria for evaluations bonds

For evaluating a bond the base informations are combined in a systematic process of aggregation according to the paradigm of the hierarchy of subjective categories. Naturally this hierarchy can be expanded easily. For simplicity of the model it is supposed that the broker or the experienced investor solve this task in an internal process of evaluation and by means of coefficients, scoring and graphical methods. (Chart analysis).

This more or less qualified process of evaluation finally yields the expected changes of price and dividends.

Such prognosis have two advantages for the investor. First he is able to control the performance based on the hierarchy of categories. Secondly he gets the opportunity to manifest his structure of expectations in a normal process. Hence he becomes aware of it and he can control and correct it.

Before we can express the method numerically we have to operationalize the criteria. They have to coincide concerning their dimensionality and have to have at least the quality of an interval scale because additions have to be performed so that they can serve as the base of an LP. So we chose monetary units for prices and dividends as these are used on the stock exchange too.

The categories are symbolized as follows :

| | |
|---|---|
| $i$ | index denoting the bond |
| ${}_{E}K_i$ | expected price |
| ${}_{R}K_i$ | risk |
| ${}_{C}K_i$ | opportunity |
| ${}_{E}D_i$ | expected dividend |
| ${}_{R}D_i$ | risk of dividend (lowest) |
| ${}_{C}D_i$ | opportunity of dividend (range) |

Symbols for informations concerning the prices and dividends :

| | |
|---|---|
| $k_i$ | price at the beginning of the actual period |
| $\underline{\bar{k}}_i$ | estimated price at the end of the period |
| $\underline{k}_i$ | estimated lowest price during the period |
| $\bar{k}_i$ | estimated highest price during the period |
| $d_i$ | last dividend |
| $\underline{\bar{d}}_i$ | next dividend |
| $\underline{d}_i$ | estimated lowest dividend |
| $\bar{d}_i$ | estimated highest dividend |

The period is fixed to one year because in Germany the dividend is paid once a year. Then the expected price is estimate :

$$(58) \qquad {}_{E}K_i = \underline{\bar{k}}_i$$

Next the two aspects of uncertainty, risk and opportunity, are considered. An investor with a high preference for certainty tries to enlarge the capital with the essential restriction, that the possible rate of loss is as small as possible. He would prefer a small but certain profit to a large but uncertain one. The risk of loss is fomulatized by the estimated lowest price during the period

$$(59) \qquad {}_RK_i = k_i - \underline{k}_i$$

The opposite of the risk of loss is the opportunity of profit. The aggressive investor will try to gain a considerable profit. If the expected risk of loss is less or equal to the possible profit, he won't buy. But the more favorable a paper is concerning the opportunity of profit the more attractive becomes the paper. The opportunity of profit is expressed as the difference between the estimated highest price and the estimated lowest price :

$$(60) \qquad {}_CK_i = (\bar{k}_i - k_i) - (k_i - \underline{k}_i)$$

$$= \bar{k}_i + \underline{k}_i - 2k_i$$

Similar operationalizations are possible for the aspects of an acceptable dividend. The expected dividend is equal to the estimated one (as was assumed for the price):

$$(61) \qquad {}_ED_i = \bar{\underline{d}}_i$$

The risk of the dividends can be described by the difference between the estimated lowest dividend and the rate of inflation (converted into DM) :

$$(62) \qquad {}_{R}D_i = \underline{d}_i - I$$

Because the rate of inflation is constant for all possible investments it can be omitted without loss of adequancy of the model. So the risk of the dividends can be represented by the estimated lowest dividends.

$$(63) \qquad {}_{R}D_i = \underline{d}_i$$

The opportunity could be operationalized analogously taken the rate of inflation into account :

$$(64) \qquad {}_{C}D_i = \bar{d}_i + \underline{d}_i - 2\,I$$

The rate of inflation again is a global constant. The lowest dividend is bounded from below by zero and therefore has a smaller range than the highest dividend. So it will correlate with the opportunity of the price. Thus it makes more sense to represent the opportunity of the dividend by the difference between the estimated highest and lowest dividend :

$$(65) \qquad {}_{C}D_i = \bar{d}_i - \underline{d}_i$$

This is a measure of uncertainty. A defensive investor will favour a lowest price as high as possible together with a small uncertainty; a speculative investor will prefer a low probability of the lowest price together with a great range above. To compare the criteria for different bonds we use a percentage scale referring to the price $K_i$.

For each share the values of the criteria are multiplied by a weight $g_i$:

(66) $\quad g_i = 100/k_i$

This percentage transformation is useful in order to make the criteria better comprehensable to the investor. It would be hard to ask for the expectations for each bond. So the expectation can be generally expressed for a category. If the interest of a bond is lower than, for instance, the rate of inflation, it is of no interest to the investor. Also the acceptable risk and the lower bounds for the opportunity can be inquired more easily if the price is supposed to be 100. The transformation to a share i is obtained by dividing by $g_i$.

The expectations of investors may be dichotomous, i.e. a share with a dividene of more than 8% is considered attractive while shares with 8% or less are not. Usually, however, this transition from "attractive" to "not attractive" is gradual. If the "rate of acceptability" is represented by values between 0 and 1, the acceptability of a share with a dividend equal to the rate of inflation might be 0 and that of a share with a dividend (in percent) twice as high as the rate of inflation is 1.

In between there is a continuum of gradual acceptance.
In between there is a continuum of gradual acceptance.

Returning to the above described paradigm of the hierarchy of subjective categories the numerical relationships between the value of the base variables and the individual acceptance are thus modelled by membership functions. For the subjective category "risk" it describes the degree of membership $\mu_R(i)$ of the alternative i to the set R of risky investments.

The individual "model" for the structure of expectance of an investor need not to be totally isomorphic to the system of categories, but it should be possible to project it into the formulated system such that the investor sees his interests represented well enough.

### III.4.1.2 The normative model

The classical formulation of decision theory distinguishes

1. a set of possible activities (decision variables),
2. a set of restrictions to bound the space of alternatives (elements within the solution space have a degree of membership equal to 1, else equal to 0),
3. a goal function which associates a degree of desirability with each feasible solution.

If the variables are additive this yields the following formal structure :

(67)
$$\begin{aligned} \max\ c^T x &= z \\ \text{s.th. } Ax &\leq b \\ x &\geq 0 \end{aligned}$$

Here x is the decision variable, in the above described problem the quantity of each bond. The matrix A contains the information of the brokers, the vectors c and b represent the goals and re strictions of the investor respectively.

In the classical LP (Problem depicted in ( ) thers is only one goal function. The different conceptions of the investors concerning the weights of the raise in price and the dividend cannot be formulated as a goal but only as restrictions. The two components of profit (price, dividend) are used as the objective function:

(68)
$$\max \sum_j (\overline{\underline{k}}_j - k_j + d_j)x_j$$

The restrictions can be derived from the operationalization of the criteria :

1. raise in price :

(69a)
$$\sum_j (\overline{\underline{k}}_j - k_j)x_j \geq b_1$$

2. risk of price :

(69b)
$$\sum_j (\underline{k}_i - k_j)x_j \geq b_2$$

3. opportunity of price :

(69c) $$\sum_j (\bar{k}_j + \underline{k}_j - 2k_j)x_j \geq b_3$$

4. profit of dividend :

(69d) $$\sum_j \underline{\bar{d}}_j x_j \geq b_4$$

5. risk of dividend (lowest) :

(69e) $$\sum \underline{d}_j x_j \geq b_5$$

6. opportunity of dividend (range) :

(69f) $$\sum_j (\bar{d}_j - \underline{d}_j)x_j \leq b_6$$

7. total budget :

(69g) $$\sum_j k_j x_j \leq b_7$$

8. portfolio condition :

(69h) $$0 \leq x_j \leq b_8 \cdot \underline{k}_j/k_j^2$$

aggregation of $k_j x_j \leq b_8 \cdot \underline{k}_j/k_j$ and non-negativity $x_j \geq 0$

Different preferences of the investors are expressed by different numerical values of $b_i$. Bascically the type of inequalities can also vary. Here we have formulated the most plausible model which can be modified if necessary when the experimental restrictions are available.

### III.4.1.3 The prescriptive model

Using the approach of Fuzzy Sets (FS) the elements of the space of alternatives are no longer associated with a degree of membership of the set $\{0.1\}$ but of the interval $[0.1]$. The rule of association is formalized by the membership function.
The same holds for the goal function because the FS-approach does not distinguish between goal function and restrictions. (Bellman & Zadeh 1970). Hence the problem has to be reformulated, such that we are looking for an alternative which is optimal according to the membership function (including goal function and restrictions). It is normally structured as follows:

$$(70)\qquad \begin{aligned} c^T x &\lesseqqgtr z \\ Ax &\lesseqqgtr B \\ x &\gtreqqless 0 \end{aligned}$$

Matrix A is to be interpreted as a list of rowvectors which are not structurally different from $c^T$. Extending matrix A by $c^T$ we obtain the matrix $A^t$ and the vector $b^t$. Thus problem (70) can be expressed as:

$$(71)\qquad \begin{aligned} A^t x &\lesseqqgtr b^t \\ x &\geqq 0 \end{aligned}$$

($A^t$ is an m+1 xn - matrix, $b^t$ an m+1 - vector.) The i-th fuzzy restriction/goal function can be transformed to an equivalent crisp problem using the following evaluation function :

$$\mu_i(x) = \begin{cases} 1 & \Leftarrow (Bx)_i \leq b_i \\ \dfrac{b_i + \varepsilon_i - (Bx)_i}{\varepsilon_i} & \Leftarrow b_i < (Bx)_i \leq b_i + \varepsilon_i \\ 0 & \Leftarrow (Bx)_i > b_i \end{cases} \tag{72}$$

The meaning can be explained by the following figure:

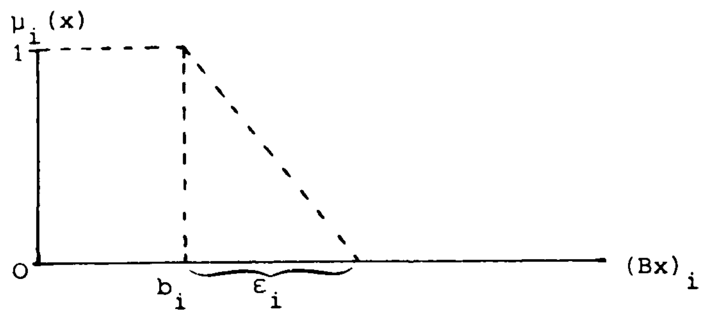



Fig. 30a : Meaning of the variables in equation (72)

The main difference to the classical LP is the variable $\varepsilon_i$ which replaces the crisp bound $b_i$ by an interval $[b_i, b_i + \varepsilon_i]$.
For each row (restriction or objective function) (72) has to be defined on the bases of the collected data. A "fuzzy decision" with the degree of membership $\mu_g(x)$ finally is a function of the aggregation of the membership functions $\mu_i(x)$. The optimal decision x is the one which maximizes the degree of membership $\mu_g(x)$.

This yields the optimizing problem :

(73) $$\max_{x \geq 0} \mu_\theta(x)$$

Using the minimum as aggregation operator as proposed by Bellman & Zadeh (1970) (73) is equivalent to

(74)
$$\begin{array}{ll|l} & \text{Max } \lambda & 0 \leq \lambda \leq 1 \\ & \text{s.th. } \lambda \leq \frac{1}{\varepsilon_i}(b_i - (Bx)_i) & i = 1(1)m+1 \\ & x \geq 0 & \end{array}$$

The model of the portfolio problem as an FLP resembles the normal LP structurally. It aims to maximize the raise in price and the dividend :

(75a) $$\text{Max} \quad \sum_j (\underline{\overline{k}}_j - k_j)x_j$$

(75b) $$\text{Max} \quad \sum_i \underline{\overline{d}}_j x_j$$

Here $x_j$ is the decision variable which denotes the quantity of bond j. The crisp bounds $b_i$ are abandoned in favour of intervals of

acceptance with lower and upper bounds ($B_i$, $B_i / i = 1, 2, \ldots, 6$), which are specified by the respective membership function.

Beside these individual categorial levels of aspirations there are two more general restrictions. First the available total budget ($B_7$) must not be exceeded. Secondly there is a maximum value ($B_8$) allowed for a single investment (portfolio condition). Here one can assume that the investor tends to allow higher investment for safer bonds. A plausible weight is the ratio of the expected lowest price to the actual price. The maximum investment for a bond can be computed as the product of the upper bound

$B_8$ with $\underline{k}_j / k_j$.

This yields the following restrictions :

1. profit in price :

(75a) $$\sum_j (\overline{k}_j - k_j) x_j \gtrsim \underline{B}_1, \overline{B}_1$$

2. risk of price :

(75b) $$\sum_j (\underline{k}_j - k_j) x_j \gtrsim \underline{B}_2, \overline{B}_2$$

3. opportunity of price :

(76c) $$\sum_j (\overline{k}_j + k_j - 2k_j) x_j \gtrsim \underline{B}_3, \overline{B}_3$$

4. profit in dividend :

(76d) $$\sum_j \overline{d}_j x_j \gtrsim \underline{B}_4, \overline{B}_4$$

5. risk of dividend (lowest) :

$$(76e) \qquad \sum_j \underline{d}_j x_j \qquad \underset{\sim}{\geq} \underline{B}_5, \ \overline{B}_5$$

6. opportunity of dividend (range) :

$$(76f) \qquad \sum_j (\overline{d}_j - \underline{d}_j) x_j \qquad \underset{\sim}{\leq} \underline{B}_6, \ \overline{B}_6$$

7. total budget :

$$(76g) \qquad \sum_j k_j x_j \qquad \underset{\sim}{\leq} B_7$$

8. portfolio condition :

$$(76h) \qquad O \geqq x_j \qquad \leq B_8 \cdot \underline{k}_j / k_j{}^2$$

(aggregation of $k_j x_j \leq B_8 \cdot \underline{k}_j / k_j$ and $x_j \geqq O$)

Now we can determine an aggregate membership function and find the combination of alternatives which maximizes this function.

Remember that the minimum operator has been choosen to model the intersection. Verbally this means that an investment is good if the increase in price and the risk of price and the opportunity of price and the increase in dividend and the risk of dividend and the opportunity of dividend are good, or more precise if the minimum over all criteria

is at a maximum.

Probably some of the investors are indifferent as to whether profit results from raise in price or from dividend. They would aggregate the subjective categories "good development of prices" and "attractive dividend by "or" which can be represented by the maximum operator for the membership functions.

Continuing our paradigm partial degrees $\gamma$ are possible between the "a" "and" and "or" aggregation. This results from the fact that some aspects of raise in price and dividend are similar while others are divergent. Thus it does not matter whether the profit results from a raise in price or from the dividend. But usually the dividend is associated with a higher certainty than the predicted raise in price which can not be timed exactly. In section 3.2 we proposed the convex combination of the intersection and the union for the aggregation in order to be able to model the individual preferences. This convex combination is formally represented in the LP by the minimum and maximum because of the better numerical tractability :

(77) $$(\bar{C}_1 - \underline{C}_1)\ \mathrm{Min} - \sum_j (\bar{\underline{k}}_j - k_j)\, x_j \leq -\underline{C}_1$$

If the investor defines lower and upper levels of aspiration for the two goals (75) , i.e. $\underline{C}_1$, $\bar{D}_1$, $\underline{C}_2$, $\bar{C}_2$ the following model can be formulated :

Goal :

(78) $$\mathrm{Max} \quad (1-\gamma) \cdot \mathrm{Min} + \gamma \cdot \mathrm{Max}$$

<u>restrictions :</u>

1. "goal I" :

(79a) $$\mu_\Theta(x) = (1-\gamma)\ \text{Min}\ [\mu_i(x)] + \gamma\ \text{Max}\ [\mu_i(x)]$$

2. "goal II" :

(79b) $$(\overline{C}_2 - \underline{C}_2)\ \text{Min} - \sum_j \underline{\overline{d}}_j \cdot x_j \leq -\underline{C}_2$$

3. profit in price :

(79c) $$(\overline{B}_1 - \underline{B}_1)\ \text{Min} - \sum_j (\underline{\overline{k}}_j - k_j)x_j \leq -\underline{B}_1$$

4. risk of price :

(79d) $$(\overline{B}_2 - \underline{B}_2)\ \text{Min} - \sum_j (\underline{k}_j - k_j)x_j \leq -\underline{B}_2$$

5. opportunity of price :

(79e) $$(\overline{B}_3 - \underline{B}_3)\ \text{Min} - \sum_j (\overline{k}_j + \underline{k}_j - 2k_j)x_j \leq -\underline{B}_3$$

6. profit of dividend :

(79f) $$(\overline{B}_4 - \underline{B}_4)\ \text{Min} - \sum_j \underline{\overline{d}}_j x_j \leq -\underline{B}_4$$

7. risk of dividend lowest) :

(79g) $$(\overline{B}_5 - \underline{B}_5)\ \text{Min} - \sum_j \underline{d}_j x_j \leq -\underline{B}_5$$

8. opportunity of dividend (ranges) :

(79h) $$(\bar{B}_6 - B_6)\ \text{Min} + \sum_j (\bar{d}_i - \underline{d}_j) x_j \le \bar{B}_6$$

9. "goal I" :

(80a) $$(\bar{C}_1 - \underline{C}_1)\ \text{Max} - \sum_j (\bar{k}_j - k_j) x_j - MY_1 \le \underline{C}_1$$

10. "goal II" :

(80b) $$(\bar{C}_2 - \underline{C}_2)\ \text{Max} - \sum_j \bar{d}_j \cdot x_j - MY_2 \le \underline{C}_2$$

11. profit in price :

(80c) $$(\bar{B}_1 - \underline{B}_1)\ \text{Max} - \sum_j (\bar{k}_j - k_j) x_j - MY_3 \le \underline{B}_1$$

12. risk of price :

(80d) $$(\bar{B}_2 - \underline{B}_2)\ \text{Max} - \sum_j (\underline{k}_j - k_j) x_j - MY_4 \le \underline{B}_2$$

13. opportunity of price :

(80e) $$(\bar{B}_3 - \underline{B}_3)\ \text{Max} - \sum_j (\bar{k}_j + \underline{k}_j - 2k_j) x_j - MY_5 \le -\underline{B}_3$$

14. profit of dividend :

(80f) $$\bar{B}_4 - \underline{B}_4)\ \text{Max} - \sum_j \bar{d}_j x_j - MY_6 \le -\underline{B}_4$$

15. risk of dividend (lowest) :

(80g) $$(\bar{B}_5 - \underline{B}_5)\ \text{Max} - \sum_j \underline{d}_j x_j - MY_7 \le -\underline{B}_5$$

16. opportunity of dividend :

(80h) $$(\bar{B}_6 - \underline{B}_6)\ \text{Max} + \sum_j (\bar{d}_j - \underline{d}_j)x_j - MY_8 \leq \bar{B}_6$$

17. exclusion

(80i) $$Y_1,\ Y_2,\ Y_3,\ Y_4,\ Y_5,\ Y_6,\ Y_7,\ Y_8 \in \{0,1\}$$

(80i') $$\text{Min}\ (Y_i) = 0$$

18. artificial variable :

(80j) $$M = \infty$$

19. total budget :

(81a) $$\sum_j k_j x_j \leq B_7$$

20. portfolio condition :

(81b) $$0 \leq x_j \leq B_8 \cdot \underline{k}_j / k_j^2$$

Restriction (79) concerns the minimum and restriction (80) the maximum. The budget restriction and the portfolio restriction are generally valid.

The $\gamma$-values for the degree of compensation of categories should be kept variable both within the hierarchy and between the individuals in order to obtain an adequate representation of the human decision. At the present state of knowledge this is only possible in the descriptive model. In the long run progress can be expected for the prescriptive model also.

### III.4.2 Preliminary study : Membership function of the investor

To get an idea about the shape of the membership function (type B), which represents the investor's attitude concerning the categories of the hierarchy of evaluation, 10 persons were interviewed, who possessed a depot of bonds. This procedure was not intended to obtain a representative random sample of all possible individual membership functions, but to

a) show, that the goals and restrictions of investors can be represented by membership functions,
b) select characteristic constellations of membership functions which can serve the brokers as an indication of the expectations of investors in the main study.

The membership functions of subject 4 are given in the following figures.

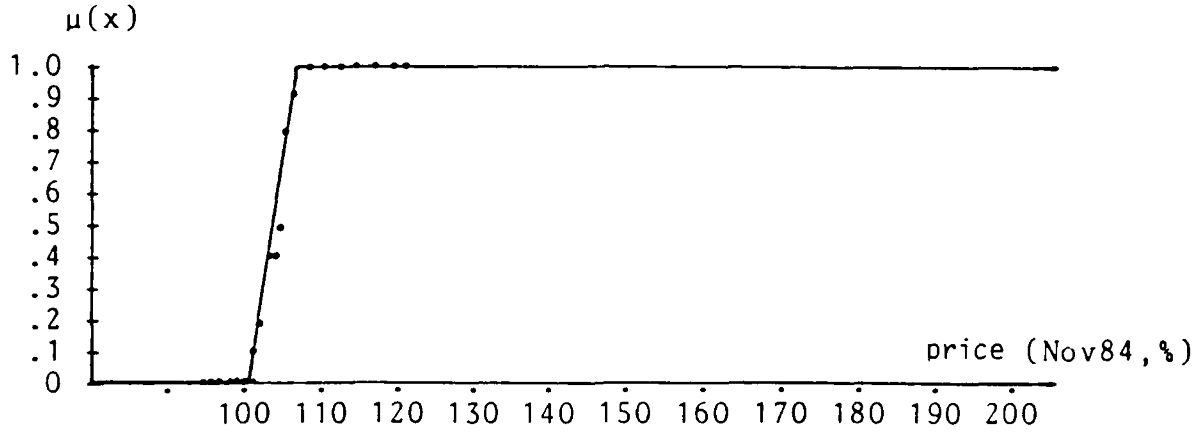



Fig. 31 investor 4, expected price

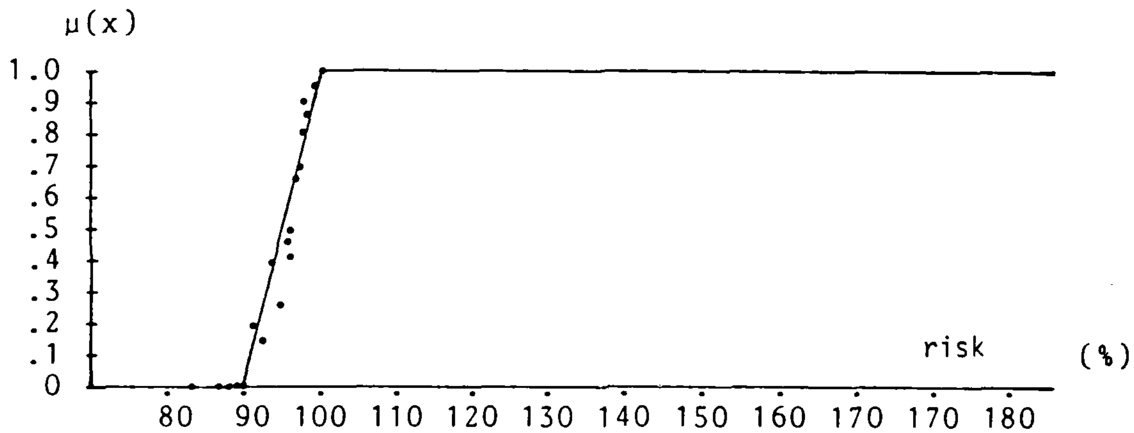



Fig.32 investor 4, risk

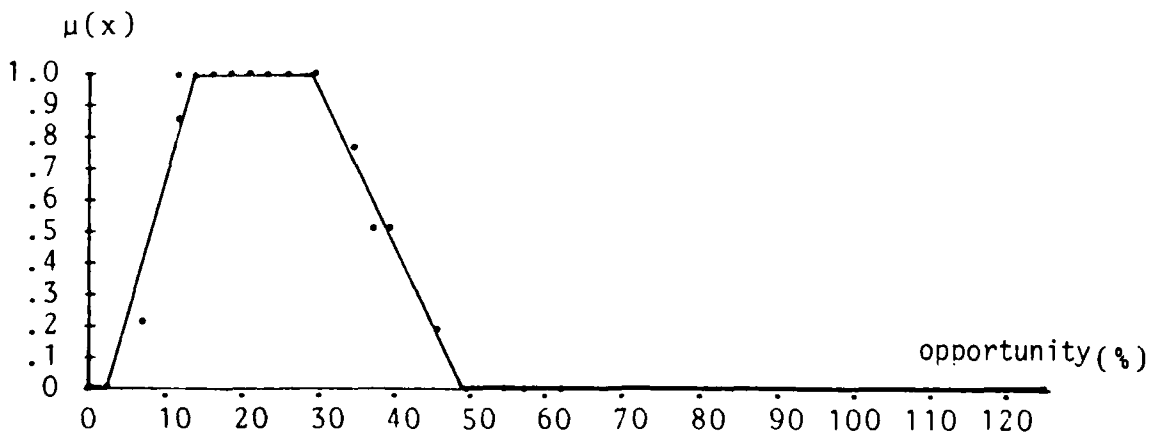



Fig.33 Investor 4, opportunity

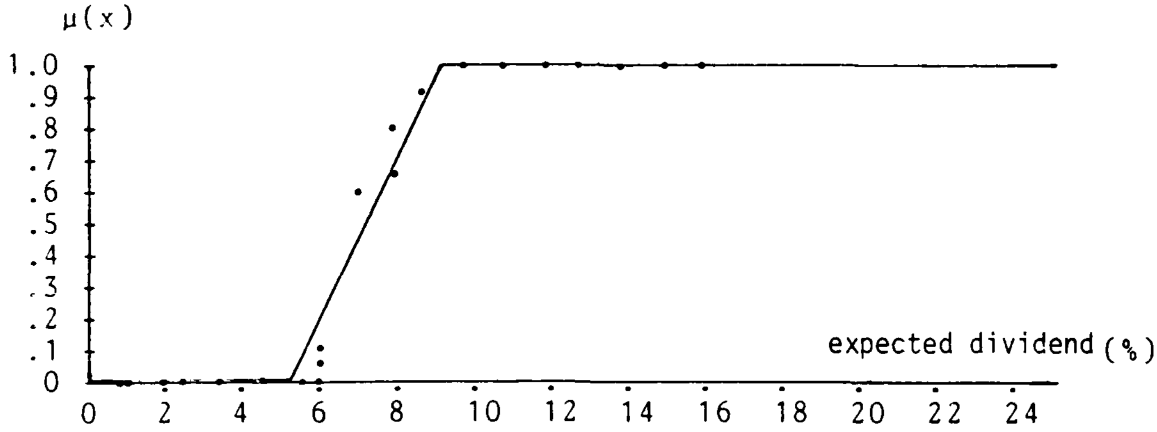



Fig. 34 investor 4, next dividend

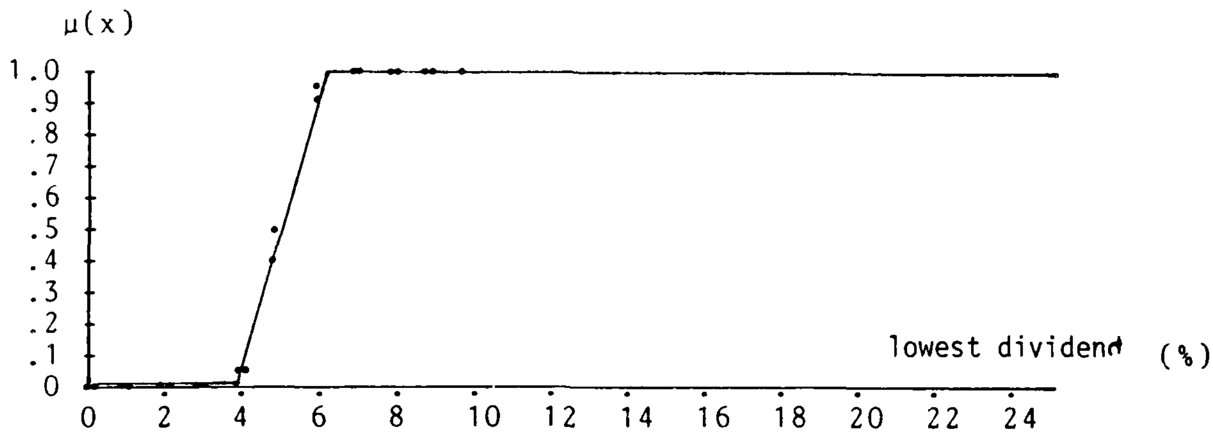



Fig. 35 investor 4, lowest

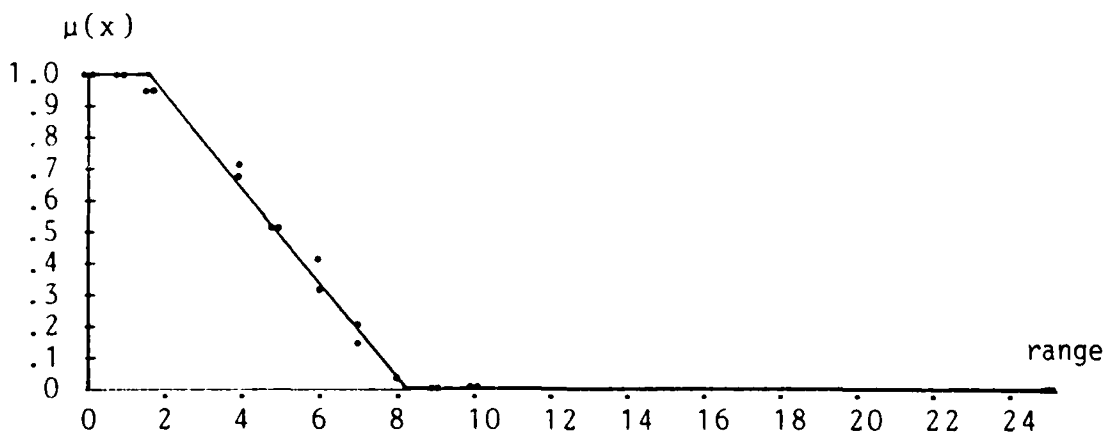



Fig. 36 investor 4, range

III.4.3 Main study : comparison of performances.

After the model had been formulated and the empirical conditions (structure of the goals and restrictions of the investors) been ensured the final and most important phase of empirical testing could be started. Based on the forecasts of the brokers concerning the development of prices and dividends of a representative selection of bonds portfolios could be determined by means of LP and FLP. These portfolios had to be compared with the propositions given by the brokers.

On the basis of the model it could be expected, that the FLP supplies "better" results than the LP because more information is considered via the membership function. ^uch a comparison, however, yields a relative judgement only; nothing can be said about whether the available informations have been used advantageously. For this purpose we would need a level of comparison which represents the general development of the stock exchange.

As a simple and plausible orientation the percentage of profit obtained by a random selection one may use :

(82)
$$v_{zuf} = 100 \frac{\Sigma \underline{\bar{k}}_z - \Sigma k_z}{\Sigma k_z}$$

$$= 100 \; (\Sigma \underline{\bar{k}}_z / \Sigma k_z - 1)$$

Here $k_z$ is the price at time t and $\underline{\bar{k}}_t$ the price at a later time t' . Index z indicates that the prices are a random representations of all bonds.

This value can be compared with the percentage of profit obtained from the bonds i of portfolio proposed by the brokers j

(83) $$v_{w(j)} = 100 \frac{\Sigma \bar{k}_i - B}{B}$$

$$= 100 \ (\Sigma \bar{k}_i / B - 1)$$

B denotes the total budget at time t . If $V_{WP(j)}$ is greater than $V_{Zuf}$ the broker has provided a good forecast. The analogous holds for the forecast based on LP and FLP.

III.4.3.1 Hypothesis

If the models contain relevant information then it can be expected that the LP yields a better portfolio than pure chance and FLP yields better results than LP : $V_{LP} < V_{FLP}$. If non-linear membership functions ($\wedge$ P) could also be intergrated the appropriate solution would probably dominate the above mentioned solutions.

Naturally the current project can only make statements about some of these relationships. The zero hypothesis assumes that neither the formal models nor the brokers contain relevant informations beyond the actual prices. It was assumed that the rates of change are all equivalent :

$$H_o: \quad V_{Zuf} = V_{WP(j)} = V_{LP(j)} = V_{FLP(j)}$$

The alternative hypothesis has to make a statement about the order of the performance of the model compared to that of the brokers. It can be expected , however, that at least some of the experts have not sufficiently considered the restrictions given by the investors (defensive, speculative ), so that they may produce an infeasible solution.

If the brokers j has been conscious about the preferences of the investor when selecting his portfolio the alternative hypothesis reads as follows :

$$H_1: \quad V_{Zuf} < V_{WP(j)} < V_{LP(j)} < V_{FLP(j)}$$

### III.4.3.2 Experimental design

The empirical part of the research aimed at obtaining from the brokers firstly estimates about the development of some selected German bonds and secondly a portfolio for the defensive and speculative investor descibed in the above chapter. In order to keep the experimental and financial effort within reasonable limits the two following restrictions were made :

1) The budget is fixed at DM 100.000.-
2) 30 bonds which are traded on German stock exchanges have been selected such that

- the different lines of trade are represented equally,
- the papers are approximately normally distributed with respect to opportunities of prices and dividends,
- the main shares (18), federal loans with fixed interests (4), real estate funds (4) and precious metals (4) are represented.

Each participant obtained a booklet of 20 pages which stated the aims of the research and which made the brokers familiar with their tasks. The completed questionnaires were returned to us by a fixed date so that we could guarantee the anonymity as well of the participants as the financial institutions.

### III.4.3.3 Evaluation and results

The test data were evaluated in the sequence of the hypotheses. The results of the four stages are summerized in figure 37 for the devensive and in figure 38 for the speculative investors. The four columns represent the results of the decision models "random", "broker" (WP), "Linear Programming" (LP) and "Fuzzy Linear Programming" (FLP). The presentation has exemplarily been limited to three brokers.

20 brokers of different financial institutions were interviewed concerning the evaluation of different German shares, real estate funds, bonds with fixed interests and precious metals. The evaluation expresses the forecasts of the expected mean, highest and

lowest prices and dividends. Based on his own forecast each participant proposed a portfolio with the budget of DM 100.000.- for the above selected defensive and speculative investors, respectively.

Our hypotheses have proved valid in all respects. Brokers are generally able to select a portfolio which yields a better expected profit than pure chance. By increasing the number of preferences and demands of the investor and the number of possible investments the probability of a feasible and even an optimal solution diminishes. By using Linear Programming both can still be obtained. Fuzzy Linear Programming enables one to find a compromise between divergent goals (high increase of prices, high dividend) in the interval between two restriction spaces and thus to enlarge the satisfaction of the user. For $\gamma = .5$ the compromise between the two goals "maximal increase in prices" and "maximal dividend" yields solutions with higher weight on the dividends. Changes to $\gamma = 1$ and $\gamma = 0$ respectively yields a higher weight for the associated components.

One conditious is, however, that the investor is able to articulate himself sufficiently or, even better, that a verbal interface can be defined which allows the freeflow of informations between the human being and the model. The proposed hierarchy of criteria, formalized using fuzzy sets, proved very useful in this respect. It does not only allow to receive informations systematically but also a to the investor. His reactions, f.i. relaxation of restrictions or specifications of certain areas of investment, could be useful for a repeated and promising analysis.

Fig. 37: Expected profits for the defensive investor resulting from the 19 portfolios based on the forecasts of the respective broker

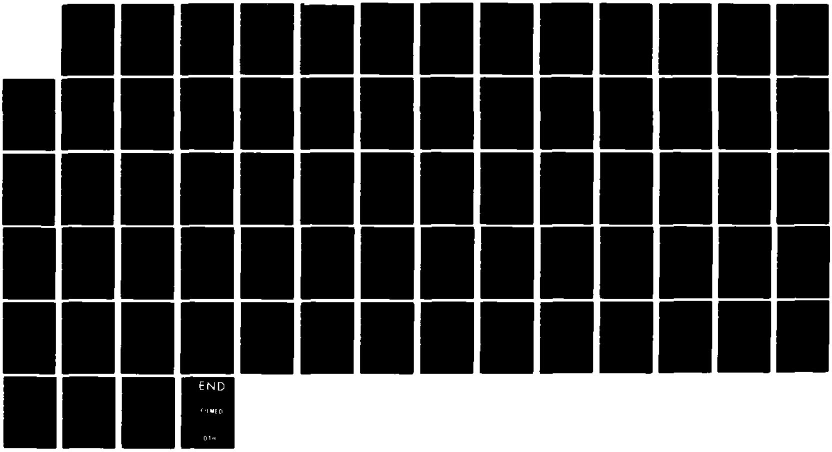
| $w_j$ / i | random: profit i.price | random: profit i.dividend | random: Σ | broker: profit i.price | broker: profit i.dividend | broker: Σ | LP: profit i.price | LP: profit i.dividend | LP: Σ | LP(r=.5): profit i.price | LP(r=.5): profit i.dividend | LP(r=.5): Σ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 7653 | 3444 | 11097 | 12800 | 5020 | 17820 | 26974 | 6000 | 32974 | [illegible] | 7600 | 29204 |
| 10 | 13823 | 3177 | 17000 | 13340 | 1277 | 14617 | 40817 | 5996 | 46813 | [illegible] | 7542 | 37214 |
| 15 | 15908 | 2935 | 18843 | 7480 | 5450 | 12930 | 28453 | 5998 | 34450 | [illegible] | 6286 | 29636 |

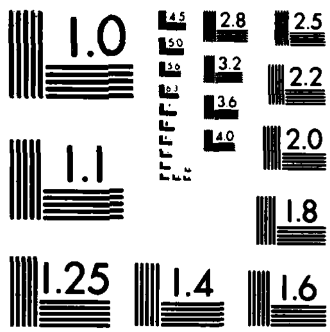
Fig. 38: Expected profits for the speculative investor resulting from the 19 portfolios based on the forecasts of the respective broker

| $w_j$ | random: profit i.price | random: profit in dividend | random: Σ | broker: profit i.price | broker: profit i.dividend | broker: Σ | LP: profit i.price | LP: profit i.dividend | LP: Σ | LP(γ=.5): profit i.price | LP(γ=.5): profit i.dividend | LP(γ=.5): Σ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 7653 | 3444 | 11097 | 18437 | 2700 | 21137 | 30696 | 4296 | 34956 | 21604 | 7599 | 29203 |
| 10 | 13823 | 3177 | 17000 | 12245 | 3560 | 15805 | 44515 | 3498 | 48013 | 35234 | 3630 | 37864 |
| 15 | 15908 | 2935 | 18843 | 35438 | 2600 | 38038 | 51160 | 1215 | 52375 | 38758 | 3392 | 42150 |

AD-A149 012 EFFICIENT ALGORITHM FOR FUZZY LINEAR PROGRAMMING WITH MULTIPLE OBJECTIVES(U) INFORM G M B H AACHEN (GERMANY F R) R HILLEKAMP ET AL. 01 DEC 84 DAJA45-82-C-0009 2/2

UNCLASSIFIED F/G 12/1 NL






MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

## III.5 EDP - Implementation

### 1) Introduction

The system "DSS" supports the decision maker solving multi-criteria problems with crisp and flexible restrictions.

The system is composed of three components :

a) Man/Machine communication

This part has the following tasks :

- to guide the decision maker through the system directed by a menu,
- to present the processed data to the decision maker and
- to allow the input and change of the data and the decision variables by the decision maker

b) Data management

This section contains the activities

- data processing and
- data update

c) System/Machine communication

This part is the interface to other software systems, which are used by "DSS". In detail this section

- generates the interface files and
- supervises and controls the execution of the software system in use.

From the above description of the components of the system it becomes obvious that the second part, "data management", depends only on the chosen programming language and therefore is unrestricted portable. Section a) "Man/Machine communication" depends on the available hard-

ware; section c) "System/Machine communication" depends on the software used.

In detail the dependance on the hardware means that communication is possible between a Hazeltine Esprit III terminal and a Cyber 175 via a synchronous line. The dependence on the software makes the application of the LP-system APEX III under NOS 1.4 necessary.

The programs of the system are coded in PASCAL and FORTRAN. The application of FORTRAN as a second language has been necessary due to the fact that on the Cyber 175 of the RWTH Aachen APEX and PASCAL are working with non-compatible data management systems.

In detail the system "DSS" processes the modules 1, 3 to 5,8 and 9 (of figure38). For this purpose two permanent libraries can be used which contain

a) all system programs as relocatable binary decks
   and
b) all data of the problem known by the system.

After starting the system the following libraries are generated depending on the needs of the user :

- libraries for intermediate results and
- storage of control procedures which are selfstarting.

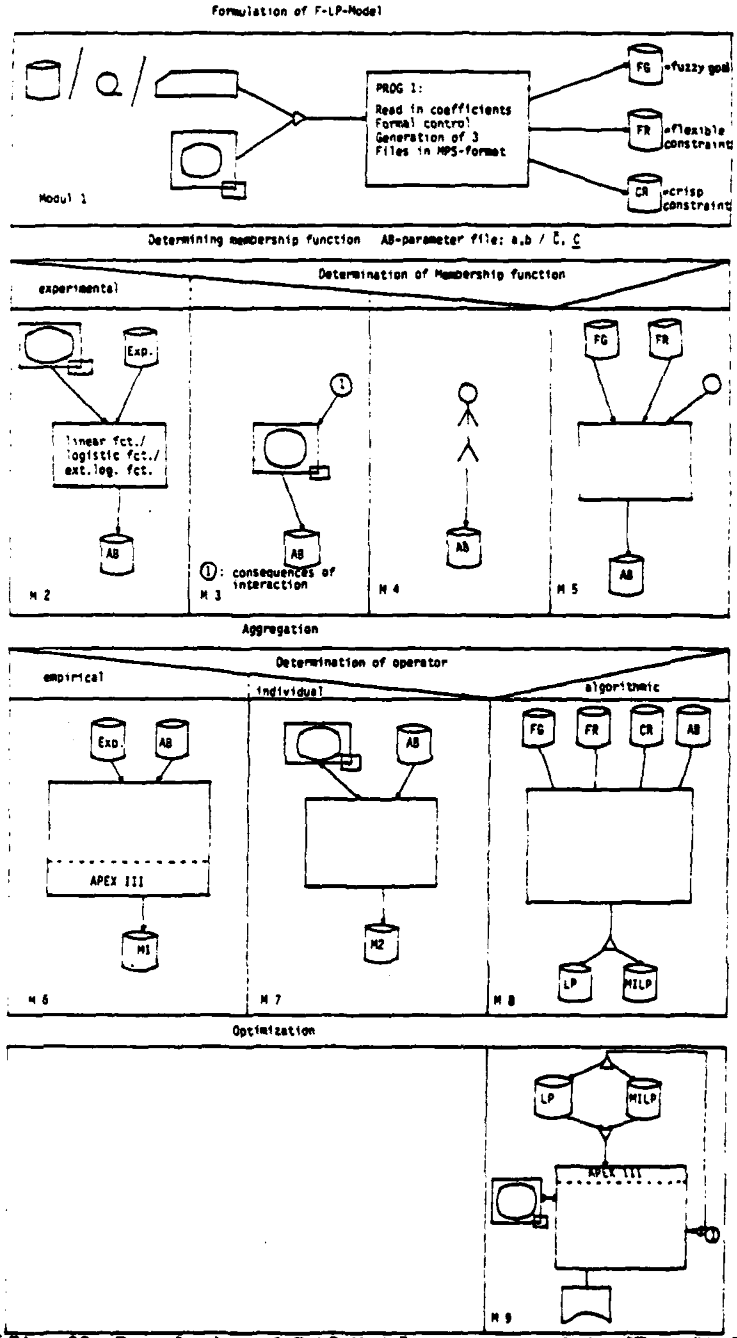



Fig. 38: Formulation of F-LP-Model

## III.6 Conclusions and Recommendations

The goals of the project as stated in the original application have been achieved: A decision support system has been designed, programmed, implemented and tested which supports decisions of the following very general kind:

1. They have to be of the "mathematical programming type", i.e. decisions have to be made which have to optimize one or more "criteria" and which are constrained by restrictions such as budgetary constraints, limited firepower, limited availability of capacities, resources or times.
2. The criteria or goals can be of different character:
   - they can be criteria which are to be strictly minimized or maximized,
   - they can represent aspiration levels which have to be achieved,
   - they can be criteria which have to be "achieved" in a more approximate way, i.e. "if possible", "as good as possible", "close to" etc.
3. The constraints can either be
   - crisp, i.e. restrictions representing well defined borders such as "at most 1 mio dollars", "at least 1000 men", etc.
   - flexible in the sense "not much more than", "basically not less than", "approximately". The reason for the constraints can be that either the data are not exactly known, the requirements are not known to the last digit, or that flexibility is desired with respect to the constraints.

For those problems the system supports the decision maker by a fuzzy multi criteria programming model interactively. The applicability of the types of membership functions and operators used in the models have been tested empirically and shown to be acceptable.

The work had to be done suject to a number of constraints:

1. Military problems could not be obtained for real testing.
2. The hardware configuration in Aachen is essentially a double Cyber 175, i.e. a very fast mainframe computer with, however, a not very comfortable and user oriented periphery. To that computer the terminals are connected via a "concentrator".
3. To solve the LP or MILP models the program APEX III was used.
4. Two years were available for all modelling, programming and testing.

Primarily due to those four constraints some improvements could not be made, which can be considered as worthwhile extensions of this project:

Empirical:

Membership functions: So far linear or transformable 2-parameter logistic membership functions have been used in the DSS. Outside the project it has been shown, however, that the 4-parameter logistic function (see fig. 7 on page 25) shows a better empirical fit, i.e. is better context adaptable. The transformation of the 2-parameter logistic function into a linear function is optimally possible via known methods. For the 4-parameter function this can only be done interatively. It would be desirable to find ways to either determine the two parameters c and d directly or to design methods to obtain optimal linear approximations as functions of a, b, c and d.

Operators: So far the min-operator, linear combinations of min and max and the "fuzzy and", represented by a linear combination of min and the algebraic mean, have been included in the DSS. It would be desirable to explore empirically how distinctions can be made between the "fuzzy and" and the "fuzzy or" when modelling a problem.

User-interface: Because of the lack of military problems the appropriateness of the model-user-interface could only be tested and improved for the portfolio problem. It would be desirable to test and possibly improve the interface in other contexts.

Modelling:

The simultaneous use of the "fuzzy and" and the "fuzzy or" or the sole use of the latter leads to integer derived models and to some other complications. One way out would be to use "hierarchical aggregation [see Werners 1984, pp. 207-213]. This would also allow the decision maker to develop his model in successive steps. To integrate this into the DSS would require more theoretical, programming and testing effort.
So far we have assumed, that the original problems did not have any integer requirements. To widen the scope of application of the DSS it would be desirable to include new approaches [f.i. Zimmermann, Pollatscheck 1984] to this end. The integration of the 4-parameter logistic function would also require additional modelling effort.

Coding:

Two improvements with respect to turn around times (i.e. waiting time of the user) could be envisaged:
In Aachen a new operating system will be installed in the near future which allows parallel processing. Then a number of processing activities could be performed in parallel rather than successively, which would reduce the waiting time of the user.

It could also be conceived a configuration in which an intelligent terminal and a mainframe share the work thus arriving at a multi-stage system which would probably provide similar improvements.

It was already mentioned that the hardware available in Aachen lacks some of the user orientation which, for instance, IBM or DEC computers do provide.
Hence a modification of the DSS for other computer periphery could considerably improve the user orientation and the portability of the system.

## III.7 Cited Literature


Aĉzel, J., Vorlesungen über Funktionalgleichungen und ihre Anwendungen, Basel-Stuttgart (Birkhäuser) 1961

Bellman, R., Zadeh, L.A., Decision Making in a Fuzzy Environment, Mgmt. Sc. 17(1970) 141-164

Campbell, N.R., Symposium: Measurement and its importance for philosophy, Aristotelian Soc. Suppl, Vol 17, 1938

Fung, L.W., Fu, K.S., An axiomatic approach to rational decision making in a fuzzy environment, in: L.A. Zadeh, K.S. Fu, K. Tanaka, M. Shimura (eds.), Fuzzy Sets and their Applications to Cognitive and Decision Processes, New York (Academic Press) 1975, 227-256

Goguen, J.A., The logic of inexact concepts, Synthese 19 (1969) 325-373

Hamacher, H., Leberling, H., Zimmermann, H.-J., Sensitivity Analysis in Fuzzy Linear Programming, Fuzzy Sets and Systems 1, (1978) 269-281

Helson, H., Adaption-level theory, New York (Harper & Blow) 1964

Hersh, H.M., Caramazza, A.A., A fuzzy set approach to modifiers and vagueness in natural language, J. Exp. Psychol. (General) 105 (1976) 254-276

Hirota, K., Concepts of probabilistic sets, Fuzzy Sets and Systems 5 (1981) 31-46

Kochen, M., Badre, A.N., On the precision of adjectives which denote fuzzy sets, Working Paper, 1973

Kuhn, H.W., Tucker, A.W., Nonlinear Programming, in: Neyman, J. (ed.) Proc. of the 2nd Berkeley Symp. on Math. Stat. and Prob. 1951

Leberling, H., On finding compromise solutions in multicriteria problems using the fuzzy min-operator, Fuzzy Sets and Systems 6 (1981) 105-118

Norwich, M., Turksen, I.B., A model for the measurement of membership and the consequences of its empirical implementation, Fuzzy Sets and Systems 12 (1984) 1-25

Rödder, W., Zimmermann, H.-J., Duality in Fuzzy Linear Programming, in: Fiacco, A.V., Kortanek, K.O. (eds.), Extremal Methods and Systems Analysis, New York (1980) 415-429

Schwab, K.-D., Ein auf dem Konzept der unscharfen Mengen basierendes Entscheidungsmodell bei mehrfacher Zielsetzung, Frankfurt (Peter Lang) 1983

Starr, M.K., Zeleny, M. (eds.), Multiple criteria decision making, Amsterdam 1977

Thole, U., Zimmermann, H.-J., Zysno, P., On the suitability of minimum and maximum operators for the intersection of fuzzy sets, Fuzzy Sets and Systems 2 (1979) 167-180

Werners, B., Interaktive Entscheidungsunterstützung durch ein flexibles mathematisches Programmierungssystem, München (Minerva) 1984

Wiedey, G., Zimmermann, H.-J., Media Selection and Fuzzy Linear Programming, J. Op. Res. Soc. 29 (1978) 1071-1084

Zadeh, L.A., Fuzzy Sets, Information & Control 8 (1965) 338-353

Zadeh, L.A., Quantitative fuzzy semantics, Information Science 3 (1971) 159-176

Zimmermann, H.-J., Fuzzy Programming and linear programming with several objective functions, Fuzzy Sets and Systems 1 (1978) 45-55

Zimmermann, H.-J., Testability and Meaning of Mathematical Models in Social Sciences, Mathematical Modelling (1980) 123-139

Zimmermann, H.-J., Pollatschek, M.A., Fuzzy 0-1 Programs, in: Zimmermann et al., Fuzzy Sets and Decision Analysis; Amsterdam New York (1984) 133-146

Zimmermann, H.-J., Zysno, P., Latent connectives in human decision making, Fuzzy Sets and Systems 4 (1980) 137-151

Zimmermann, H.-J., Zysno, P., Decisions and evaluations by hierarchical aggregation of information, Fuzzy Sets and Systems 10 (1983) 243-260

Zimmermann, H.-J., Zysno, P., Zugehörigkeitsfunktionen: Modellierung, empirische Bestimmung und Verwendung in Entscheidungsmodellen, DFG Forschungsbericht 1984

Zysno, P., The integration of concepts within judgmental and evaluative processes, in: R. Trappl (ed.), Progress in Cybernetics and Systems Research, Vol. 8, Washington D.C. (Hemisphere Publ.) 1980

## IV Documentation of the System

"DSS" is a program system for supporting the decision maker solving problems with several objectives and crisp and fuzzy (flexible) restrictions The system fulfills the following tasks:

- storage and maintenance of different data of the problem
- determination of membership functions and extremal solutions
- aggregation of membership functions and computation of compromise alternatives
- presentation of further local information and processing of interactive modifications.

This documentation aims at making the reader familiar with the way the system is working. Therefore first a dialog with the system will be presented examplarily. Then the existing data files and programs will be documented. Finally a complete representation of error messages and screen masks will be given.

The documentation is structured as follows:

1. Structure of the system
2. Description of the data files
   2.1 Survey
   2.2 Detailed descriptions
      2.2.1 PROBES
      2.2.2 PROBDA
      2.2.3 PROBL
      2.2.4 APEX-.. data fi s
      2.2.5 TAPE4

3. Documentation of the programs
    3.1 MAINTEN1
    3.2 CREAT
    3.3 DELET
    3.4 DESCR
    3.5 UPDTE
    3.6 EXT
    3.7 PROBSOL
    3.8 BINGLP
    3.9 INDVLP
    3.10 COMPL
    3.11 SOLUT
    3.12 MAINCCL
4. Error Messages
5. Masks for Dialog

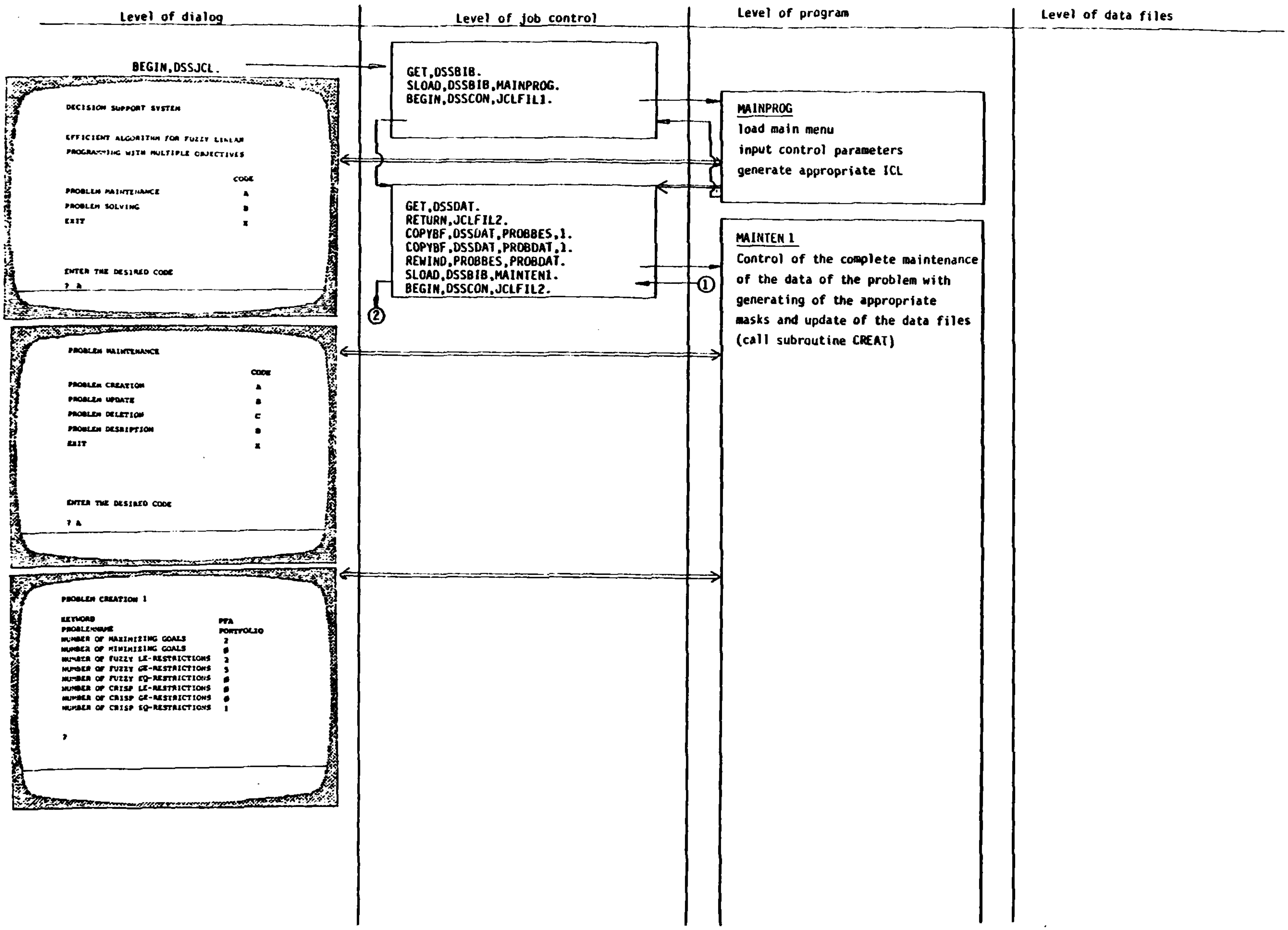
## IV.1 Structure of the system

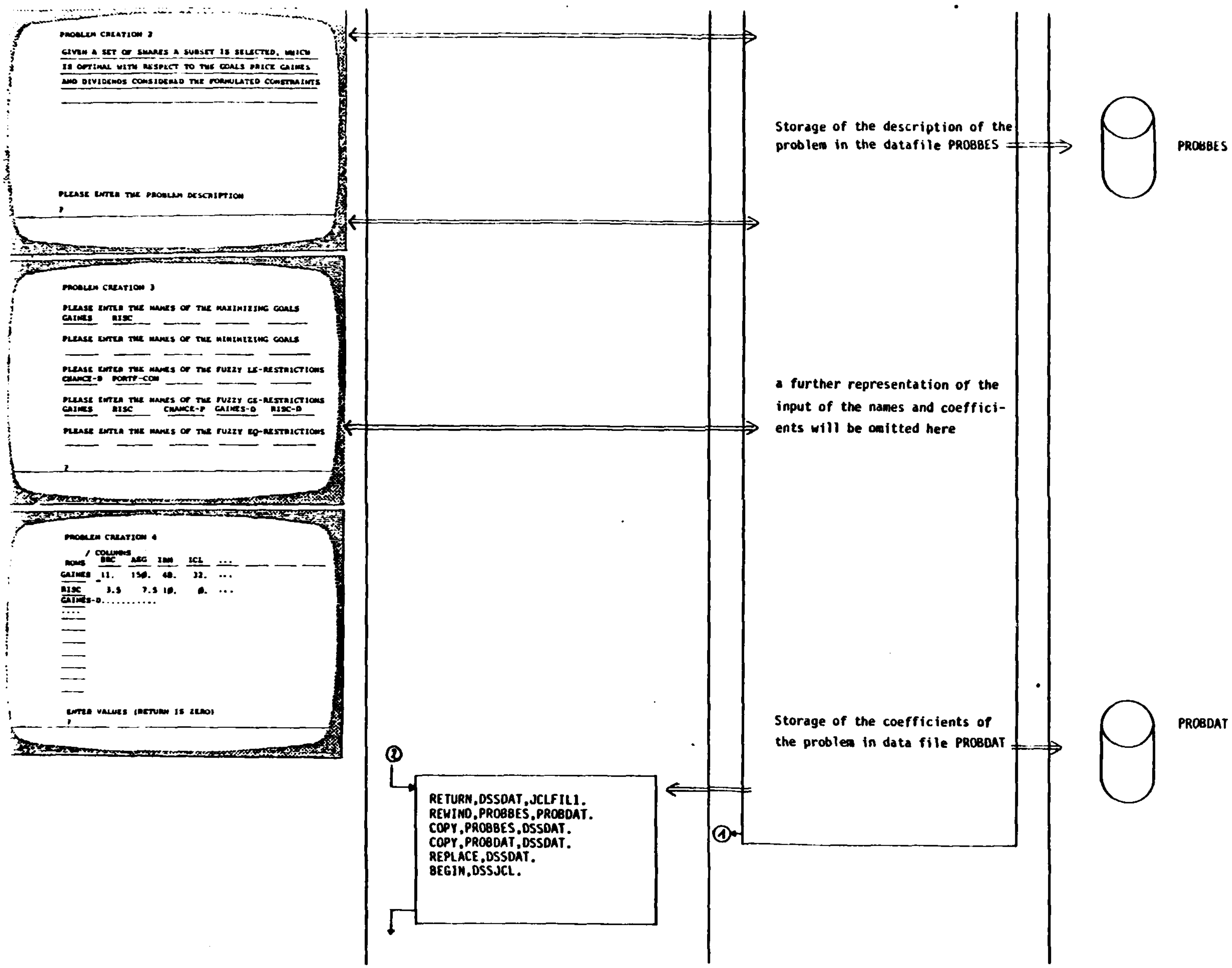
The system "DSS" is a menu-oriented dialog system for solving multi-criteria problems with crisp and fuzzy restrictions. A static description of the structure shall be omitted, because the structure of the system is highly dependent of the wishes of the decision maker. Instead we shall try to clarify the structure by describing a terminal session exemplarily. The session will be represented on four levels:

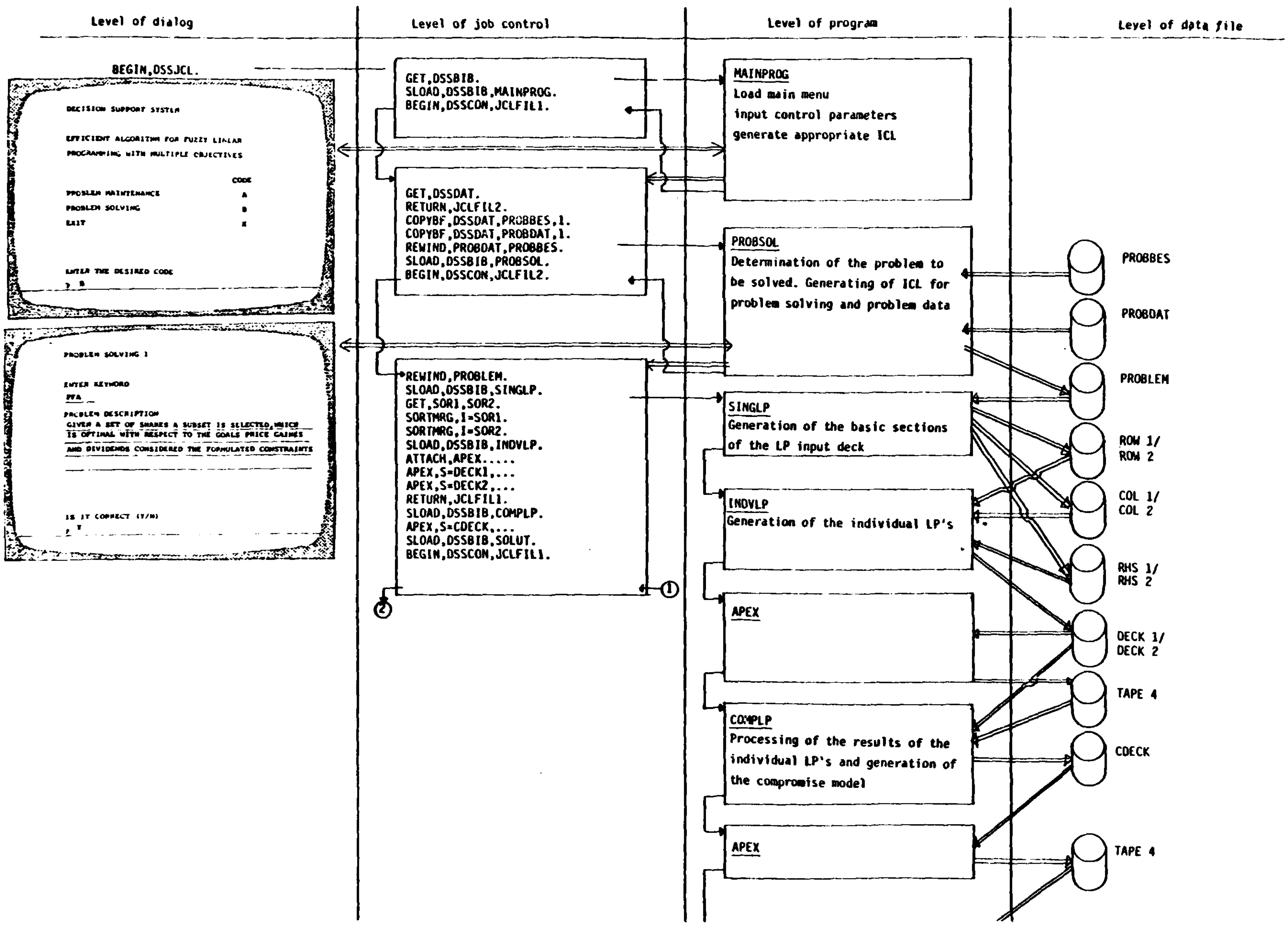
- level of dialog
- level of job control
- level of data files
- level of programs.

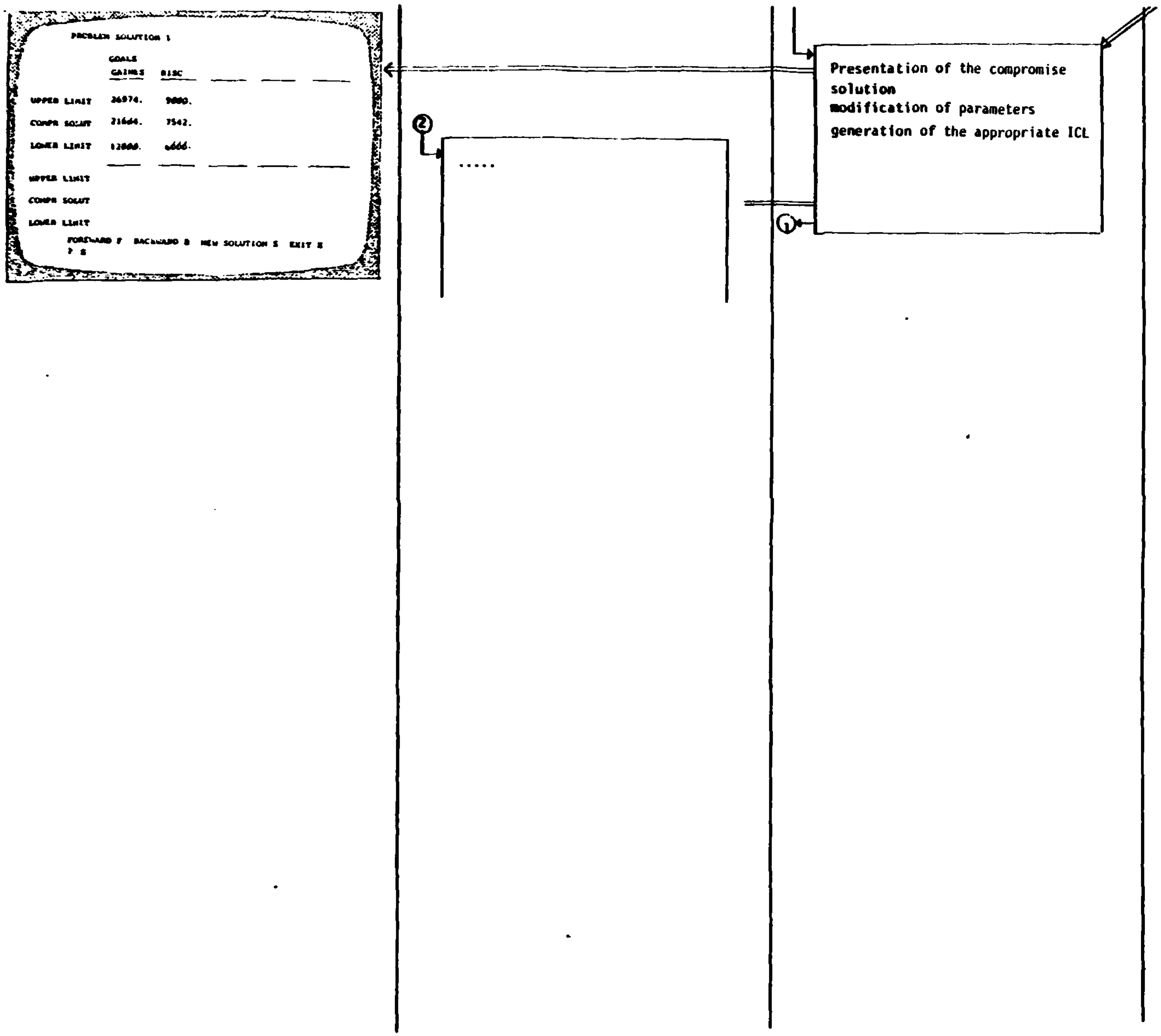
We shall represent, for instance, the input of a new problem and the solution of this problem. We will omit the complete representation of all masks for reasons of space.

In the following a single arrow denotes a flow of control, a double one a flow of data.


Level of dialog
Level of job control
Level of program
Level of data files
BEGIN,DSSJCL.
GET,DSSBIB.
SLOAD,DSSBIB,MAINPROG.
BEGIN,DSSCON,JCLFIL1.
MAINPROG
load main menu
input control parameters
generate appropriate ICL
DECISION SUPPORT SYSTEM
EFFICIENT ALGORITHM FOR FUZZY LINEAR
PROGRAMMING WITH MULTIPLE OBJECTIVES
CODE
PROBLEM MAINTENANCE A
PROBLEM SOLVING B
EXIT X
ENTER THE DESIRED CODE
? A
GET,DSSDAT.
RETURN,JCLFIL2.
COPYBF,DSSDAT,PROBBES,1.
COPYBF,DSSDAT,PROBDAT,1.
REWIND,PROBBES,PROBDAT.
SLOAD,DSSBIB,MAINTEN1.
BEGIN,DSSCON,JCLFIL2.
1
2
MAINTEN 1
Control of the complete maintenance
of the data of the problem with
generating of the appropriate
masks and update of the data files
(call subroutine CREAT)
PROBLEM MAINTENANCE
CODE
PROBLEM CREATION A
PROBLEM UPDATE B
PROBLEM DELETION C
PROBLEM DESRIPTION D
EXIT X
ENTER THE DESIRED CODE
? A
PROBLEM CREATION 1
KEYWORD PFA
PROBLEMNAME PORTFOLIO
NUMBER OF MAXIMIZING GOALS 2
NUMBER OF MINIMIZING GOALS 0
NUMBER OF FUZZY LE-RESTRICTIONS 2
NUMBER OF FUZZY GE-RESTRICTIONS 5
NUMBER OF FUZZY EQ-RESTRICTIONS 0
NUMBER OF CRISP LE-RESTRICTIONS 0
NUMBER OF CRISP GE-RESTRICTIONS 0
NUMBER OF CRISP EQ-RESTRICTIONS 1
?


PROBLEM CREATION 2
GIVEN A SET OF SHARES A SUBSET IS SELECTED, WHICH
IS OPTIMAL WITH RESPECT TO THE GOALS PRICE GAINES
AND DIVIDENDS CONSIDERAD THE FORMULATED CONSTRAINTS
PLEASE ENTER THE PROBLEM DESCRIPTION
PROBLEM CREATION 3
PLEASE ENTER THE NAMES OF THE MAXIMIZING GOALS
GAINES RISC
PLEASE ENTER THE NAMES OF THE MINIMIZING GOALS
PLEASE ENTER THE NAMES OF THE FUZZY LE-RESTRICTIONS
CHANCE-D PORTF-COM
PLEASE ENTER THE NAMES OF THE FUZZY GE-RESTRICTIONS
GAINES RISC CHANCE-P GAINES-D RISC-D
PLEASE ENTER THE NAMES OF THE FUZZY EQ-RESTRICTIONS
PROBLEM CREATION 4
ENTER VALUES (RETURN IS ZERO)
Storage of the description of the
problem in the datafile PROBBES
PROBBES
a further representation of the
input of the names and coefficients will be omitted here
Storage of the coefficients of
the problem in data file PROBDAT
PROBDAT
RETURN,DSSDAT,JCLFIL1.
REWIND,PROBBES,PROBDAT.
COPY,PROBBES,DSSDAT.
COPY,PROBDAT,DSSDAT.
REPLACE,DSSDAT.
BEGIN,DSSJCL.


Level of dialog
Level of job control
Level of program
Level of data file
BEGIN,DSSJCL.
DECISION SUPPORT SYSTEM
EFFICIENT ALGORITHM FOR FUZZY LINEAR
PROGRAMMING WITH MULTIPLE OBJECTIVES
CODE
PROBLEM MAINTENANCE A
PROBLEM SOLVING B
EXIT X
ENTER THE DESIRED CODE
? B
PROBLEM SOLVING I
ENTER KEYWORD
PFA
PROBLEM DESCRIPTION
GIVEN A SET OF SHARES A SUBSET IS SELECTED,WHICH
IS OPTIMAL WITH RESPECT TO THE GOALS PRICE GAINES
AND DIVIDENDS CONSIDERED THE FORMULATED CONSTRAINTS
IS IT CORRECT (Y/N)
? Y
GET,DSSBIB.
SLOAD,DSSBIB,MAINPROG.
BEGIN,DSSCON,JCLFIL1.
GET,DSSDAT.
RETURN,JCLFIL2.
COPYBF,DSSDAT,PROBBES,1.
COPYBF,DSSDAT,PROBDAT,1.
REWIND,PROBDAT,PROBBES.
SLOAD,DSSBIB,PROBSOL.
BEGIN,DSSCON,JCLFIL2.
REWIND,PROBLEM.
SLOAD,DSSBIB,SINGLP.
GET,SOR1,SOR2.
SORTMRG,I=SOR1.
SORTMRG,I=SOR2.
SLOAD,DSSBIB,INDVLP.
ATTACH,APEX.....
APEX,S=DECK1,...
APEX,S=DECK2,...
RETURN,JCLFIL1.
SLOAD,DSSBIB,COMPLP.
APEX,S=CDECK,...
SLOAD,DSSBIB,SOLUT.
BEGIN,DSSCON,JCLFIL1.
1
2
MAINPROG
Load main menu
input control parameters
generate appropriate ICL
PROBSOL
Determination of the problem to
be solved. Generating of ICL for
problem solving and problem data
SINGLP
Generation of the basic sections
of the LP input deck
INDVLP
Generation of the individual LP's
APEX
COMPLP
Processing of the results of the
individual LP's and generation of
the compromise model
APEX
PROBBES
PROBDAT
PROBLEM
ROW 1/
ROW 2
COL 1/
COL 2
RHS 1/
RHS 2
DECK 1/
DECK 2
TAPE 4
CDECK
TAPE 4


PROBLEM SOLUTION 1
GOALS
UPPER LIMIT
COMPR SOLUT
LOWER LIMIT
Presentation of the compromise solution
modification of parameters
generation of the appropriate ICL

## IV.2 Description of the data files

### IV.2.1 Survey

| data file / modul | MAINPROG | MAINTEN1 | PROBSOL | SINGLP | INDVLP | COMPLP | APEX | SOLUT |
|---|---|---|---|---|---|---|---|---|
| PROBBES | | PU | | | | | | |
| PROBDAT | | PU | PR | | | PR | | |
| PROBLEM | | | WW | WR | | | | PR |
| ROW1/2 | | | | WW | WR | | | |
| COL1/2 | | | | WW | WR | | | |
| RHS1/2 | | | | WW | WR | | | |
| DECK1/2 | | | | | WW | WR | WR | |
| TAPE4 | | | | | | WR | WW | WR |
| CDECK | | | | | | WW | WR | WU |

Legend:

PR - } library file { reading
PW - } library file { writing
PU - } library file { modificating

WR - } working file { reading
WW - } working file { writing
WU - } working file { modificating

## IV.2.2 Detailed descriptions

### IV.2.2.1 PROBES

General informations

| | |
|---|---|
| Name of the file: | PROBES |
| Organization: | sequential |
| Type of file: | permanent library modul |
| Record length: | ≤ 364 characters |
| Max. number of records: | according to the stored problem descriptions which have come in during the processing |
| Short description: | File contains a global short description of the present problems. |

| | | | | | | |
|---|---|---|---|---|---|---|
| structure of record | ISUCH | PROBNAME | MAX | MIN | FLE | |
| format of record | A(3) | A(20) | I2 | I2 | I2 | |
| creation in | CREAT | CREAT | CREAT | CREAT | CREAT | |
| use in | MAINTEN 1 | MAINTEN 1 | MAINTEN 1 | MAINTEN 1 | MAINTEN 1 | |
| modification in | UPDTE | UPDTE | UPDTE | UPDTE | UPDTE | |
| | FGE | FEQ | CLE | CGE | CEQ | LAENGE |
| | I2 | I2 | I2 | I2 | I2 | I(5) |
| | CREAT | CREAT | CREAT | CREAT | CREAT | CREAT |
| | MAINTEN 1 | MAINTEN 1 | MAINTEN 1 | MAINTEN 1 | MAINTEN 1 | MAINTEN 1 |
| | UPDTE | UPDTE | UPDTE | UPDTE | UPDTE | UPDTE |
| | BESCH | | | | | |
| | A(320) | | | | | |
| | CREAT | | | | | |
| | MAINTEN 1 | | | | | |
| | UPDTE | | | | | |

| | | |
|---|---|---|
| ISUCH | - | 3 digit alphanumerical searching key for determination of the problem to be processed |
| PROBNAME | - | 20 digit alphanumerical name of the problem |
| MAX | - | 2 digit numerical number of maximizing objectives |
| MIN | - | " " " " " minimizing objectives |
| FLE | - | " " " " " fuzzy ≤ --restrictions |
| FGE | - | " " " " " fuzzy ≥ - restrictions |
| FEQ | - | " " " " " fuzzy - -resctrictions |
| CLE | - | " " " " " crisp ≤ - restrictions |
| CGE | - | " " " " " crisp ≥ - restrictions |
| CEQ | - | " " " " " crisp - - restrictions |
| LAENGE | - | 5 digit numerical description of the length of problem data in records |
| BESCH | - | contains a short formulation of the actual problem (up to 320 alphanumerical signs) |

## IV.2.2.2 PROBDAT

General information:

| | |
|---|---|
| Name of the file: | PROBDAT |
| Organization: | sequential |
| Type of file: | permanent multifile library modul |
| Record length: | ≤ 34 characters |
| Max. number of records: | summation of the contents of the array LAENGE in the file "PROBES" |
| Short description: | file contains the initial data of all stored problems |

| | | | | | |
|---|---|---|---|---|---|
| structure of record | RCTYP | RCNAME | RCNAMI | RANBON | |
| format of record | I2 | A(20) | A(10) | I1 | |
| creation in | CREAT | CREAT | CREAT | CREAT | |
| use in | MAINTEN1/PROBSOL | | | | |
| Modification in | UPDTE | UPDTE | UPDTE | UPDTE | |

Redefinition

| | | | | | |
|---|---|---|---|---|---|
| RCTYP | VARANZ | BLO | B | BUP | |
| I2 | I2 | F10.4 | F10.4 | F10.4 | |
| CREAT | CREAT | CREAT | CREAT | CREAT | |
| | | | | | |
| UPDTE | UPDTE | UPDTE | UPDTE | UPDTE | |

Redefinition

| | | | | | |
|---|---|---|---|---|---|
| RCTYP | VARNO | VWERT | | | |
| I2 | I2 | F10.4 | | | |
| CREAT | CREAT | CREAT | | | |
| | | | | | |
| UPDTE | UPDTE | UPDTE | | | |

RCTYP - type of the following records

1 - maximizing objective

2 - minimizing objective

3 - fuzzy ≤/≥ - restriction

4 - fuzzy = - restriction

5 - crisp ≤/≥ - restriction

6 - crisp = - restrictions

7 - variable without bounds

8 - variable with bounds

10 - row description

20 - input of coefficients

RCNAME - user name of rows and columns

RCNAMI - internal name of rows and columns

RANBON - characterization whether variable is bounded

VARANZ - number of NNE in a row

BLO - values of the right hand side BLO ≤ B ≤ BUP for fuzzy restrictions;

B - for fixed restrictions only B is given

BUP -

VARNO - number of a variable in a row

VWERT - value of coefficient

### IV.2.2.3 Problem

General information:

| | |
|---|---|
| Name of the file: | PROBLEM |
| Organization: | index-sequential |
| Type of file: | working |
| Record length: | variable |
| Max. number of records: | according to the number of functions of the problem to be solved |
| Short description: | File contains processed initial problem for generating the single LP/MIP model formulations |

Structure of data record

| words: | 1 | 2 | 3 . . . n |
|---|---|---|---|
| | Integer | Integer | variable section |
| | type of record | record length | |

Type of record:
1 characterization of the problem
2 max - objective
3 min - objective
4 fuzzy objective ≤
5 fuzzy objective ≥
6 fuzzy objective =
7 crisp objective ≤
8 crisp objective ≥
9 crisp objective =

Type 1:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| # max OF | # min OF | # FR ≤ | # FR ≥ | # FR = | # CR ≤ | # CR ≥ | # CR = |

Type 2 + 3:

| | |
|---|---|
| column index | value |

Type 4 + 5:

| | | | | |
|---|---|---|---|---|
| $\underline{b}$ | $\bar{b}$ | column index | value | . . . |

Type 6:

| | | | | | |
|---|---|---|---|---|---|
| $\underline{b}$ | b | $\bar{b}$ | column index | value | . . . |

Type 7 - 9:

| | | | |
|---|---|---|---|
| RHS | column index | value | . . . |

## IV.2.2.4 APEX input data files

General information

| | |
|---|---|
| Name of the file: | ROW1 / ROW2, COL1 / COL2, RHS1 / RHS2, DECK1 / DECK2, CDECK |
| Organization: | sequential |
| Type of file: | working |
| Record length: | ≤ 72 characters |
| Max. number of records: | ≤ 2* number of functions + number NNE of the coefficient matrix + 10 |
| Short description: | data files contains APEX input structures |

(For further informations see APEX III, Reference Manual, CDC, Publ.-No. 76070000, 1976)

## IV.2.2.5 TAPE 4

General information:

| | |
|---|---|
| Name of the file: | TAPE 4 |
| Organization: | sequential |
| Type of file: | working |
| Record length: | 70 characters |
| Max. number of records: | number of functions + number of variables + 3 |
| Short description: | file contains special APEX output which can be processed by FORTRAN programs |

Each solution file contains two header records, with each record being seven 60-bit words in length. The first header contains the following information:

| Word | CR Cell | Type | Description |
|---|---|---|---|
| 1 | KNPROB | Alpha | Name of problem |
| 2 | KNOBJ | Alpha | Name of objective function |
| 3 | KNRHS | Alpha | Name of right-hand side |
| 4 | KNBND | Alpha | Name of bounds set or blank |
| 5 | RPSOBJ | Real | Multiplier of objective, usually +1. or -1. |
| 6 | RPSRHS | Real | Multiplier of right-hand side, usually +1. |
| 7 | RDOBJFN | Real | Current value of objective function |

The second header contains the following information:

| Word | CR Cell | Type | Description |
|---|---|---|---|
| 1 | KNCHOBJ | Alpha | Name of change objective function or blank |
| 2 | KNCHRHS | Alpha | Name of change right-hand side or blank |
| 3 | KNRNG | Alpha | Name of ranges set or blank |
| 4 | RPCHOBJ | Real | Multiplier of change objective function or zero |
| 5 | RPCHRHS | Real | Multiplier of change right-hand side or zero |
| 6 | LJROWS | Integer | Number of rows in the problem |
| 7 | LJCOLS-LKRHS | Integer | Number of columns in the problem, excluding right-hand sides |

A seven-word record (60 bits per word) is written for each row and column (excluding right-hand sides) in the problem. All row records are described first, followed by column records.

The row detail record includes the following information:

| Word | Type | Description |
|---|---|---|
| 1 | Alpha | Name of the row |
| 2 | Real | Row activity level |
| 3† | Real | Slack activity |
| 4† | Real | Right-hand side lower limit |
| 5† | Real | Right-hand side upper limit |
| 6 | Real | Marginal value (dual) |
| 7‡ | Octal | Special packed word |

Similarly, the column detail record contains:

| Word | Type | Description |
|---|---|---|
| 1 | Alpha | Name of the column |
| 2 | Real | Activity level of the column |
| 3 | Real | Original cost (objective coefficient) |
| 4† | Real | Column lower bound |
| 5† | Real | Column upper bound |
| 6 | Real | Marginal value ($d_j$ or reduced cost) |
| 7‡ | Octal | Special packed word |

As indicated, word 7 is a special word. Its information is packed in the following form:

| Bits | Value | Description |
|---|---|---|
| 59-58 | 00 | Variable is okay |
| | 01 | Reserved for future use |
| | 10 | Variable is nonoptimal |
| | 11 | Variable is infeasible |
| | | NOTE: A sign test specifies whether the variable is nonoptimal or infeasible |
| 57-30 | 0 | Reserved for future use |
| 29-12 | — | Variable number according to input order |
| 11-10 | — | Basis status, including the following four types of status: |
| | 00 | • Nonbasic status |
| | 01 | • Nonbasic at upper bound status |
| | 10 | • Basic status |
| | 11 | • Reserved for future use |
| 9-8 | | Variable type, including the following four types of variables: COLUMN TYPE (ROW TYPE) |
| | 00 | • Fixed (E = EQUALITY) |
| | 01 | • Plus (L = LESS THAN OR EQUAL) |
| | 10 | • Minus (G = GREATER THAN OR EQUAL) |
| | 11 | • Free (N = FREE (NONCONSTRAINING)) |
| 7 | — | Upper bound indicator specifying one of the following two conditions: (The U bit)† |
| | 0 | • No upper bound |
| | 1 | • Upper bound exists |
| 6 | — | Lower bound indicator (columns only) specifying either: (The L bit)† |
| | 0 | • No lower bound (lower is zero) |
| | 1 | • Lower bound exists |
| 5-0 | — | Variable type, including one of the following types of variable: |
| | 00 | • Row variable |
| | 01 | • Column variable |
| | 02 | • Binary variable |
| | 04 | • Integer variable |
| | 05 | • Type 1 SOS variable |
| | 06 | • Type 2 SOS variable |

A final seven-word record is written to indicate the end of the special file. It takes the following form:

| Word | Type | Description |
|---|---|---|
| 1 | Alpha | $$END$$bbb (where b = blank) |
| 2-7 | — | zero |

## IV.3 Description of the program

### IV.3.1 MAINTEN 1

Name: MAINTEN 1

Task: processing of all maintenance activities for the data of the appropriate CCL procedures

Language: FORTRAN

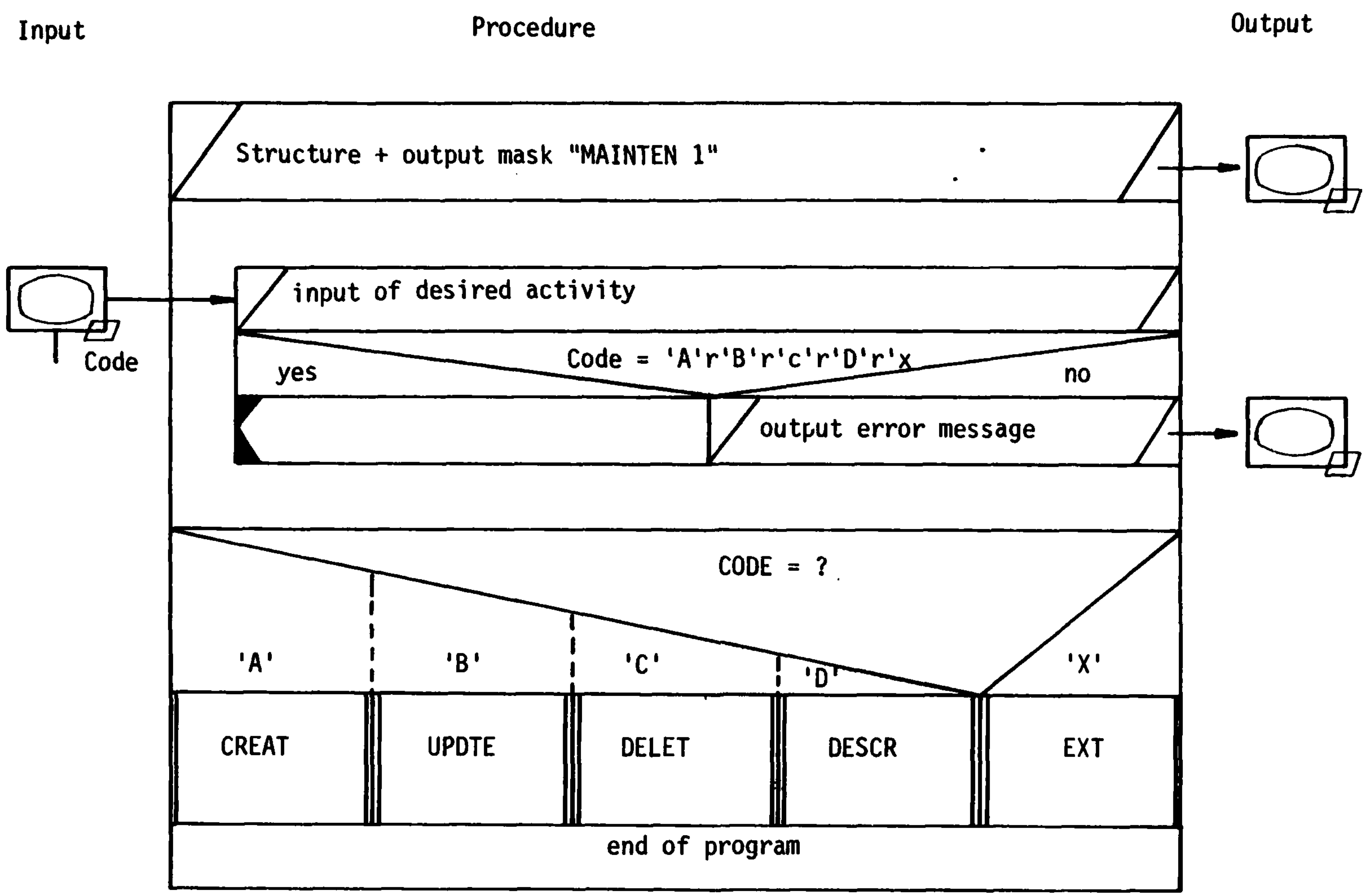
Status:
- calling programs: MAINCCL
- called programs: CREAT, DELET, DESCR, UPDTE, EXT

  control procedure

  main program

  subroutine
- mode: **dialog**

Data files: PROBDAT, PROBBES



Input
Procedure
Output
Structure + output mask "MAINTEN 1"
input of desired activity
Code
Code = 'A'r'B'r'c'r'D'r'x
yes
no
output error message
CODE = ?
'A'
'B'
'C'
'D'
'X'
CREAT
UPDTE
DELET
DESCR
EXT
end of program

## IV.3.2 CREAT

Name:

Task: input of data for problem description and problem, formal check of these data storage in the data files PROBDAT and PROBBES

Language: FORTRAN

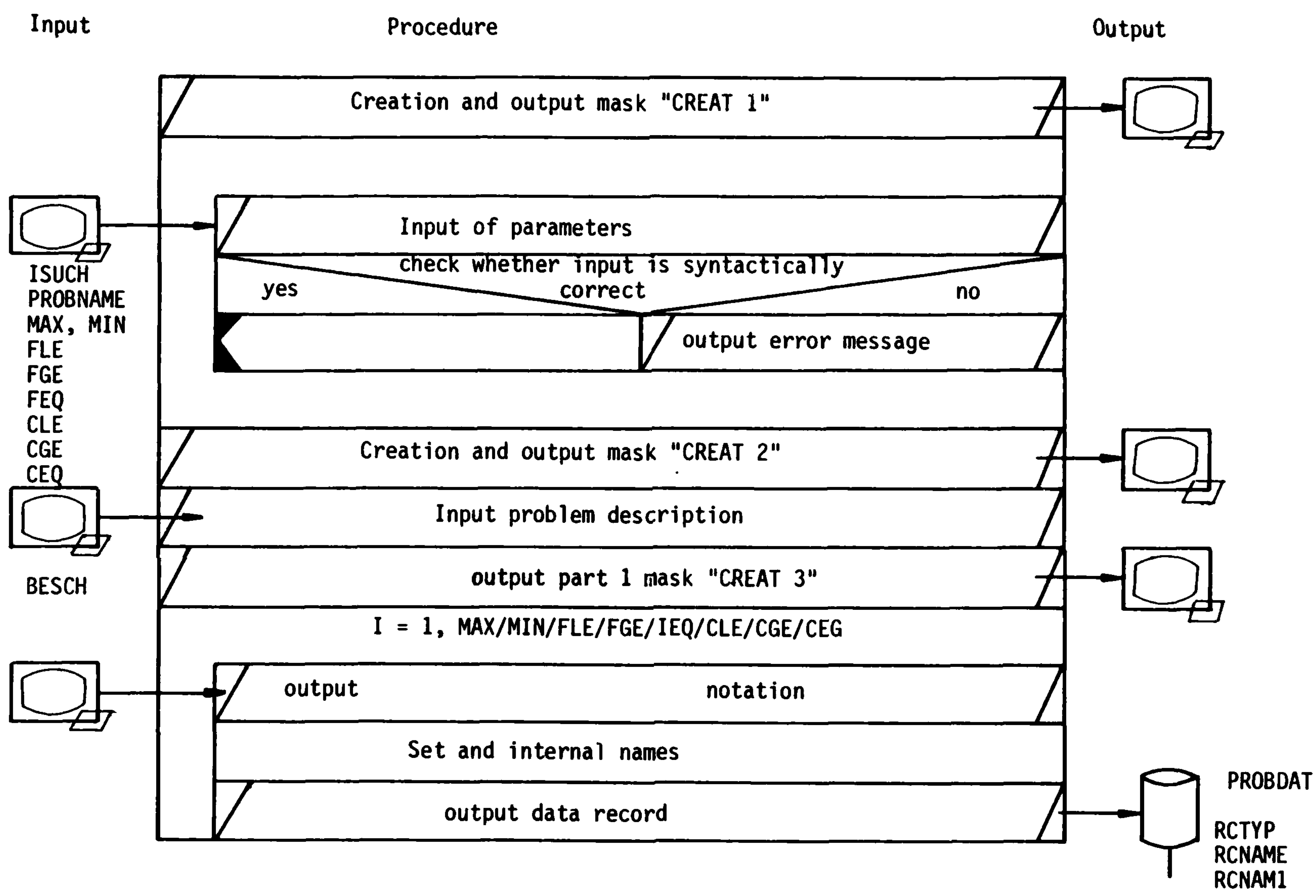
Status:
- calling programs: MAINTEN 1
- called programs: --

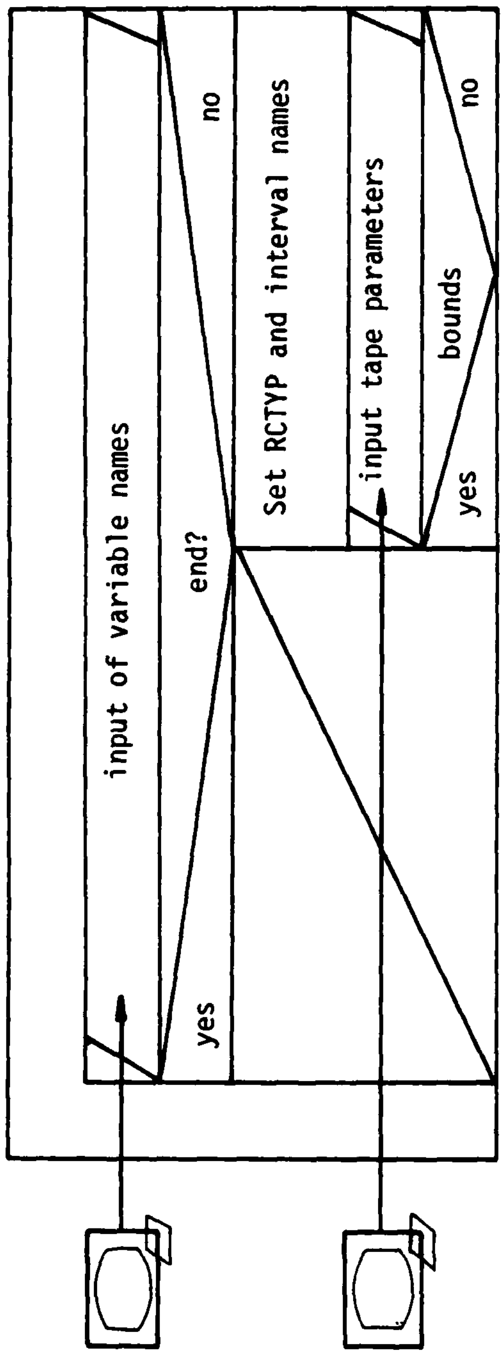
  control procedure

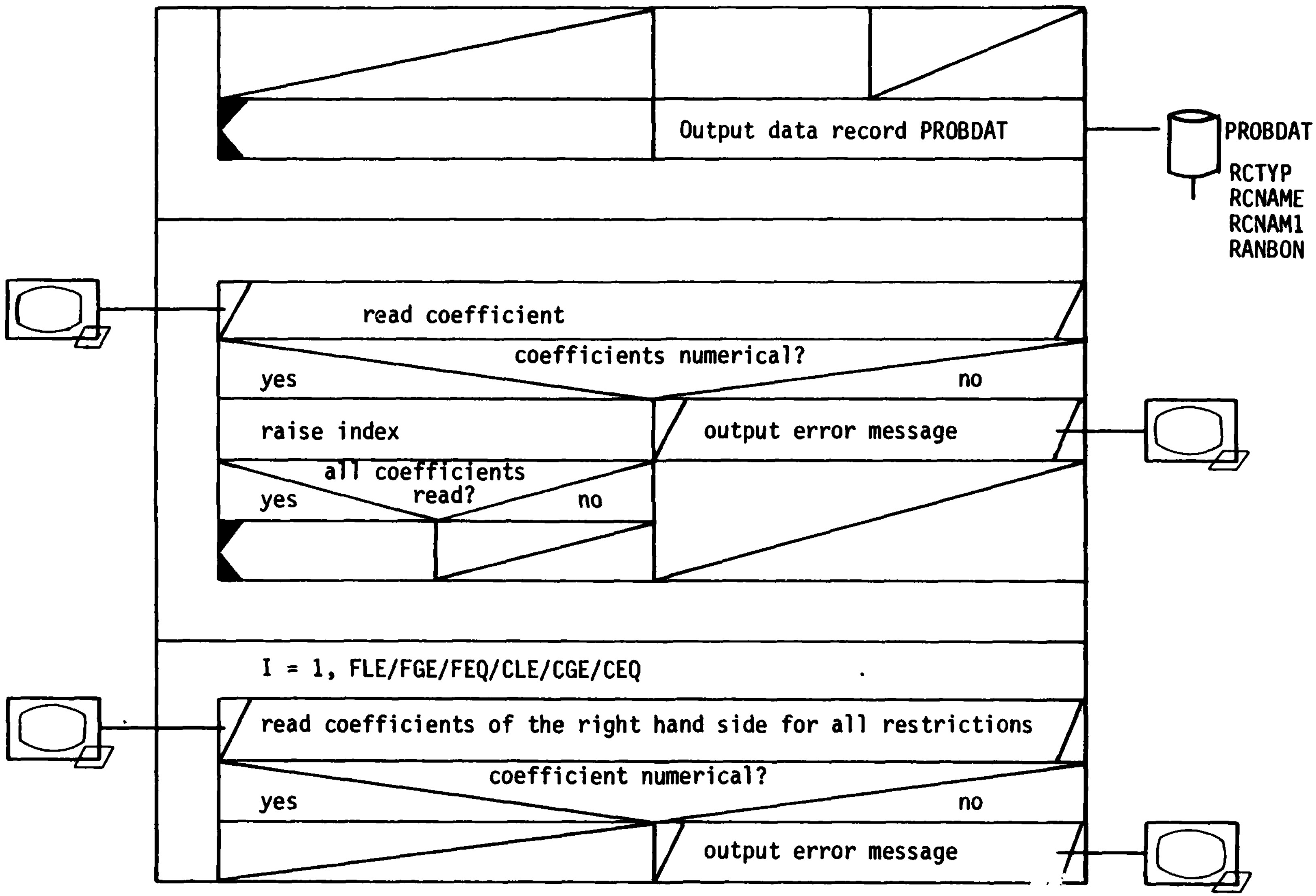
  main program

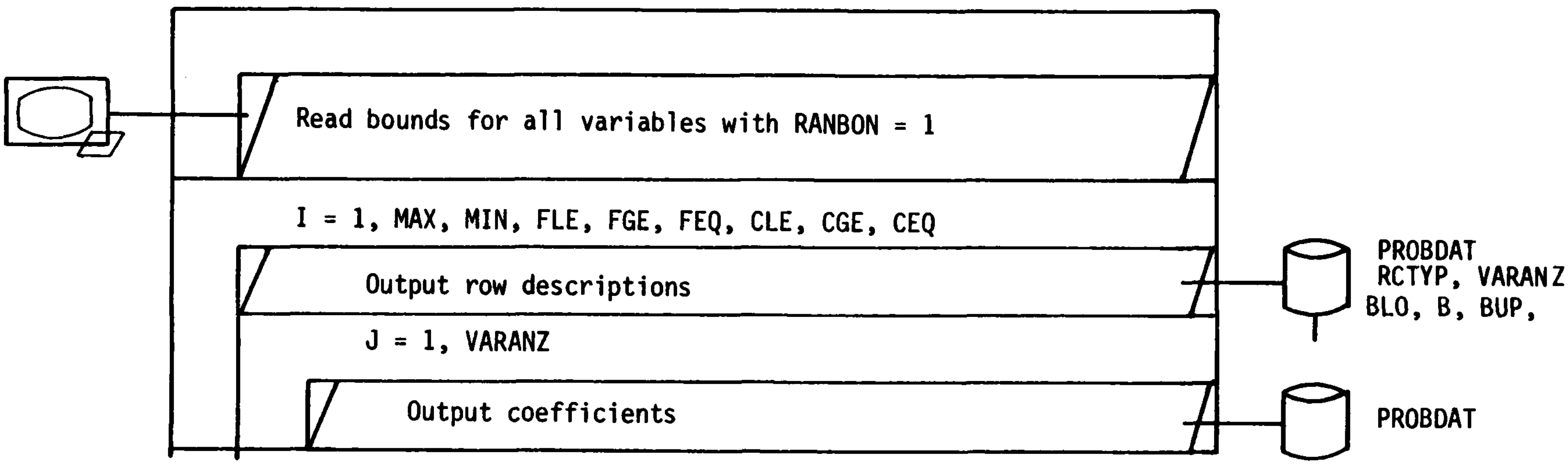
  subroutine
- mode: dialog

Data files: PROBDAT, PROBBES


Input
Procedure
Output
Creation and output mask "CREAT 1"
Input of parameters
ISUCH
PROBNAME
MAX, MIN
FLE
FGE
FEQ
CLE
CGE
CEQ
check whether input is syntactically correct
yes
no
output error message
Creation and output mask "CREAT 2"
Input problem description
BESCH
output part 1 mask "CREAT 3"
I = 1, MAX/MIN/FLE/FGE/IEQ/CLE/CGE/CEG
output
notation
Set and internal names
output data record
PROBDAT
RCTYP
RCNAME
RCNAM1


input of variable names
end?
yes
no
Set RCTYP and interval names
input tape parameters
bounds
yes
no


Output data record PROBDAT
PROBDAT
RCTYP
RCNAME
RCNAM1
RANBON
read coefficient
coefficients numerical?
yes
no
raise index
output error message
all coefficients read?
yes
no
I = 1, FLE/FGE/FEQ/CLE/CGE/CEQ
read coefficients of the right hand side for all restrictions
coefficient numerical?
yes
no
output error message


Read bounds for all variables with RANBON = 1
I = 1, MAX, MIN, FLE, FGE, FEQ, CLE, CGE, CEQ
Output row descriptions
J = 1, VARANZ
Output coefficients
PROBDAT
RCTYP, VARANZ
BLO, B, BUP,
PROBDAT

## IV.3.3 DELET

Name: DELET

Task: delete all data of a problem both in the file of description and of problem, reorganisation of these data files

Language: FORTRAN

Status: - calling programs: MAINTEN 1

- called programs: --

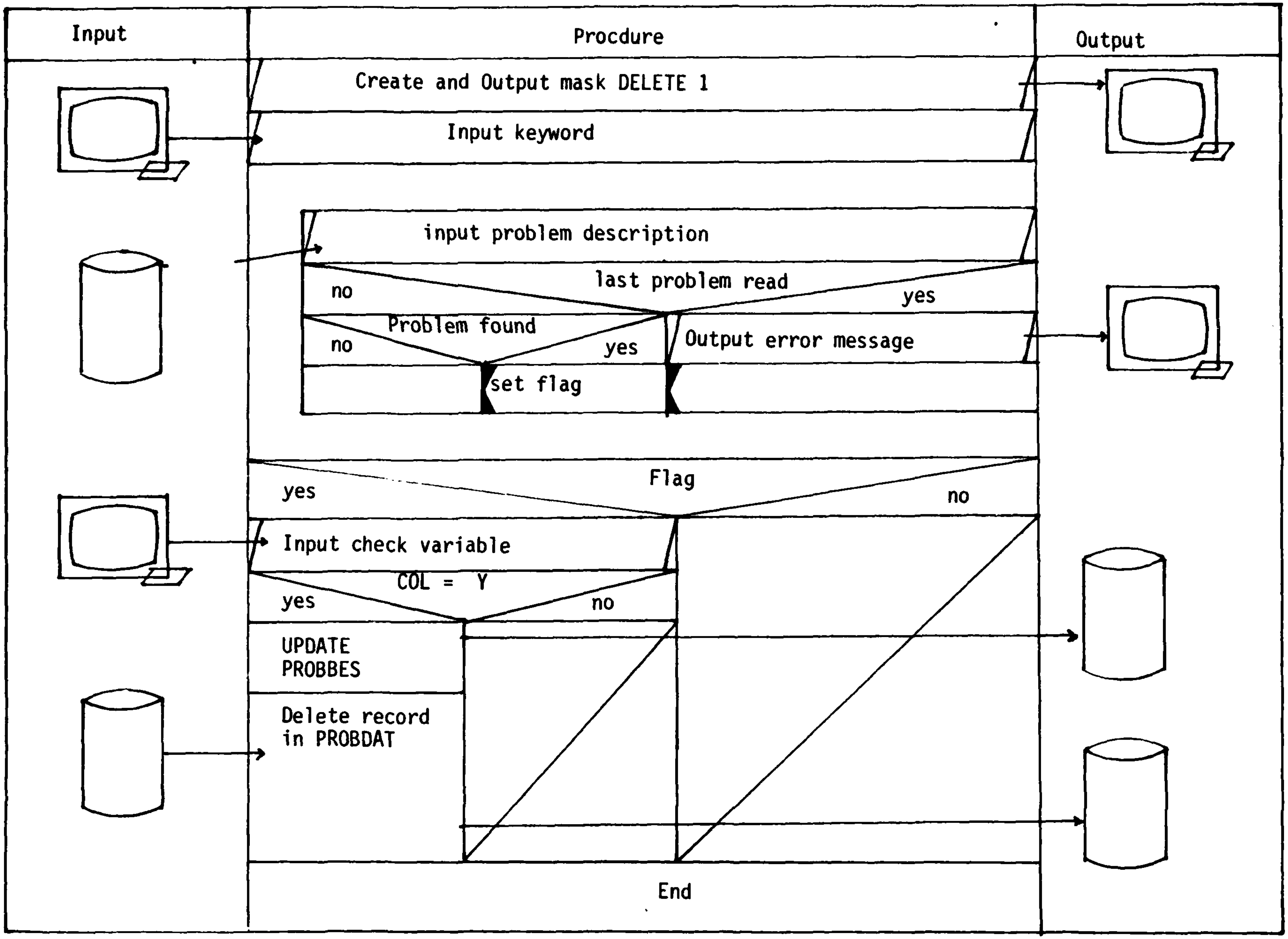
  control procedure

  main program

  soubroutine

- mode: dialog

Data files: PROBDAT, PROBBES


Input
Procdure
Output
Create and Output mask DELETE 1
Input keyword
input problem description
last problem read
no
yes
Problem found
no
yes
Output error message
set flag
Flag
yes
no
Input check variable
COL = Y
yes
no
UPDATE
PROBBES
Delete record
in PROBDAT
End

## IV.3.4 DESCR

Name: DESCR

Task: This program enables the decision maker to page in the data file of problem description and to search for a particular problem

Language: FORTRAN

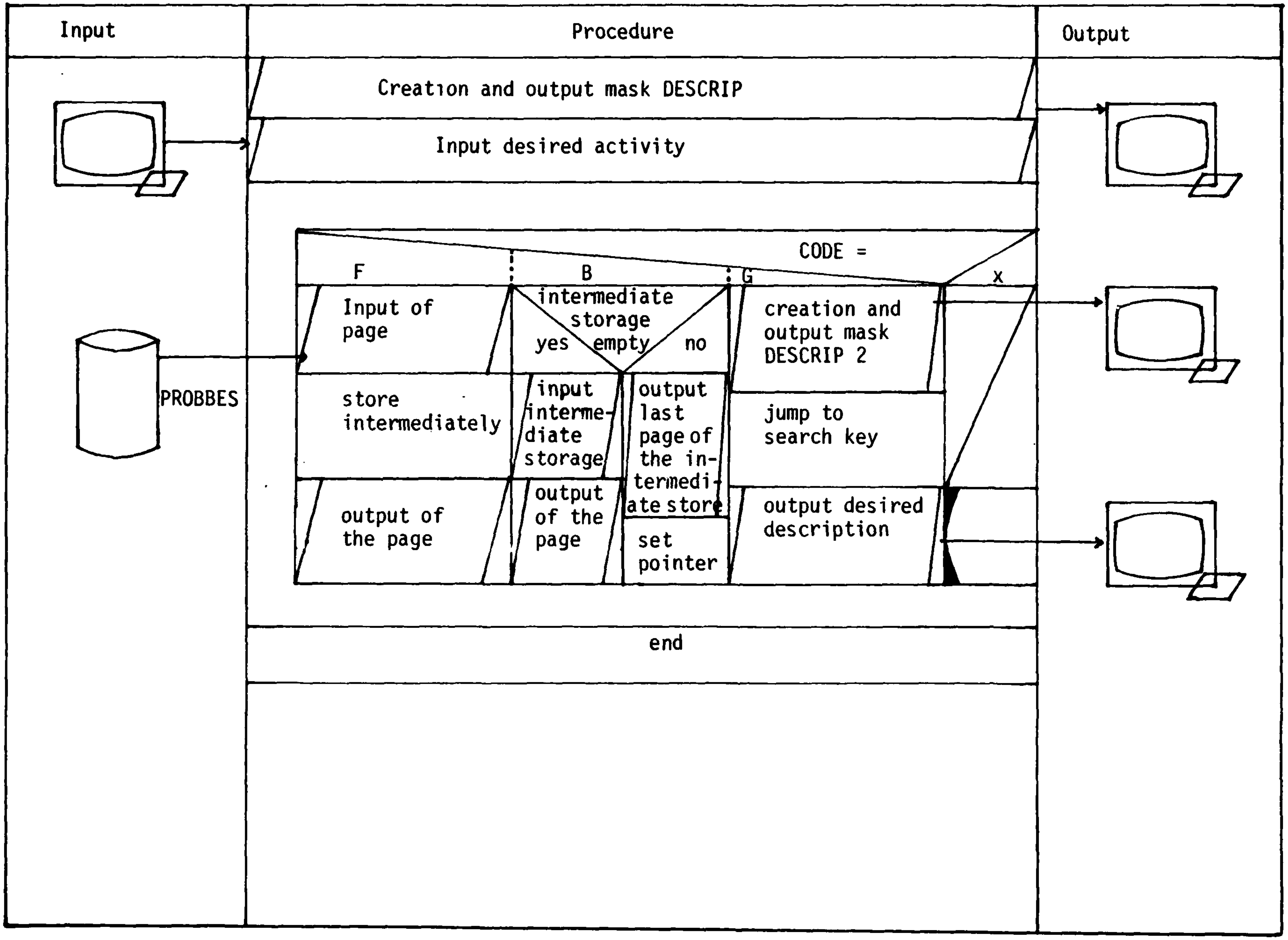
Status:
- calling programs: MAINTEN 1
- called programs: --

  control procedure

  main program

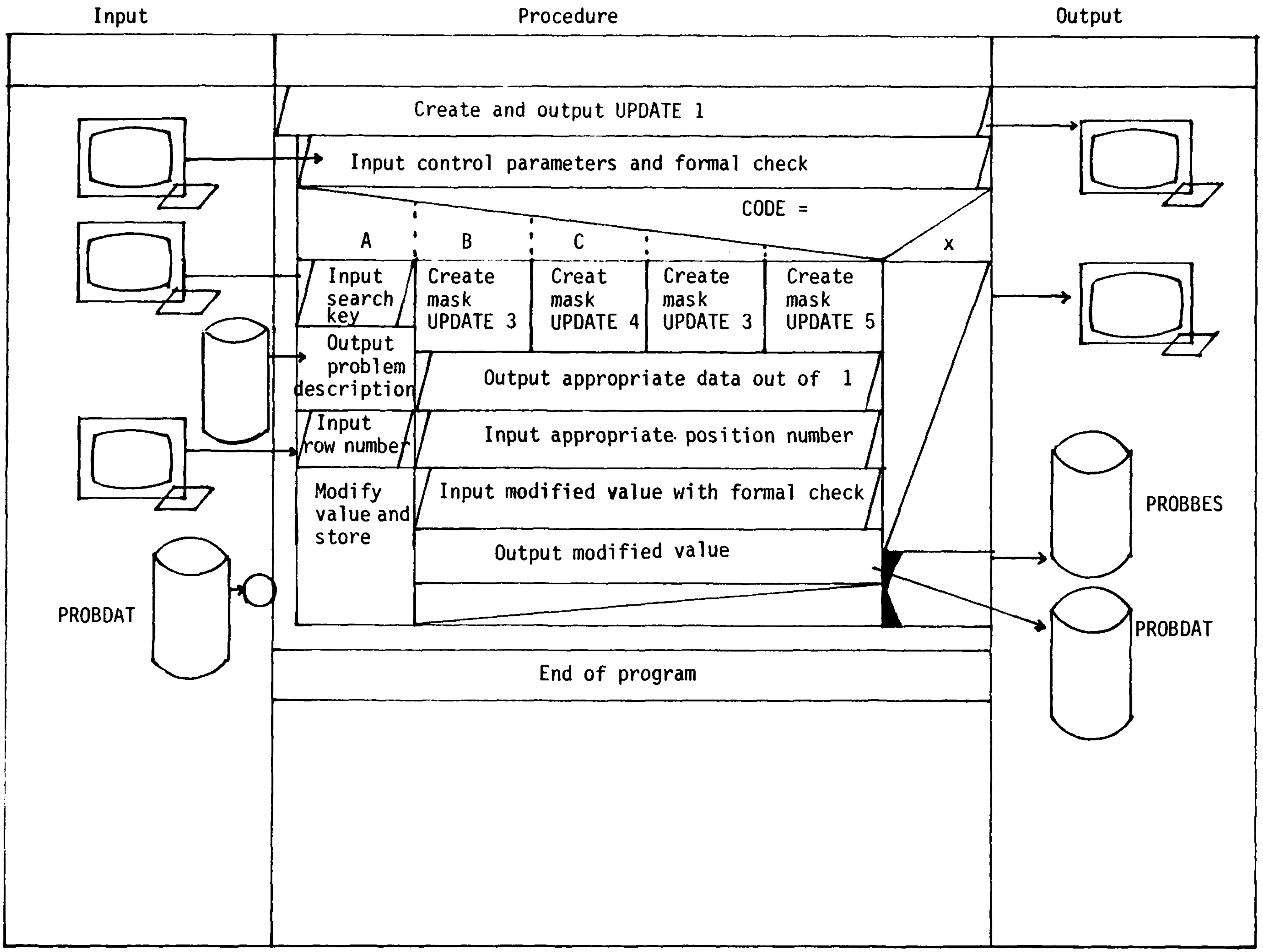
  <u>subroutine</u>
- mode: dialog

Data files: PROBDAT, PROBBES


Input
Procedure
Output
Creation and output mask DESCRIP
Input desired activity
CODE =
F
B
G
x
Input of page
store intermediately
output of the page
intermediate storage empty
yes
no
input intermediate storage
output of the page
output last page of the intermediate store
set pointer
creation and output mask DESCRIP 2
jump to search key
output desired description
PROBBES
end

## IV.3.5 UPDTE

| | |
|---|---|
| Name: | UPDTE |
| Task: | change single data of a problem in the data files PROBDAT and PROBBES |
| Language: | FORTRAN |
| Status: | - calling programs: MAINTEN 1 |
| | - called programs: -- |
| | control procedure |
| | main program |
| | <u>subroutine</u> |
| | - mode: dialog |
| Data files: | PROBDAT, PROBBES |


Input
Procedure
Output
Create and output UPDATE 1
Input control parameters and formal check
CODE =
A
B
C
x
Input search key
Create mask UPDATE 3
Creat mask UPDATE 4
Create mask UPDATE 3
Create mask UPDATE 5
Output problem description
Output appropriate data out of 1
Input row number
Input appropriate position number
Modify value and store
Input modified value with formal check
Output modified value
End of program
PROBDAT
PROBBES
PROBDAT

IV.3.6 EXT

Name: EXT

Task: according to the program activities desired in MAINTEN 1 create a job control file which now controls the system "DSS" and which returns to the main menu after termination

Language: FORTRAN

Status:
- calling programs: MAINTEN 1
- called programs: --
  control procedure
  main program
  <u>subroutine</u>
- mode: dialog

Data files: PROBDAT, PROBBES

IV.3.7 PROBSOL

Name: PROBSOL

Task: extract a data record of the library data file PROBDAT and create an index sequential data file PROBLEM for further processing

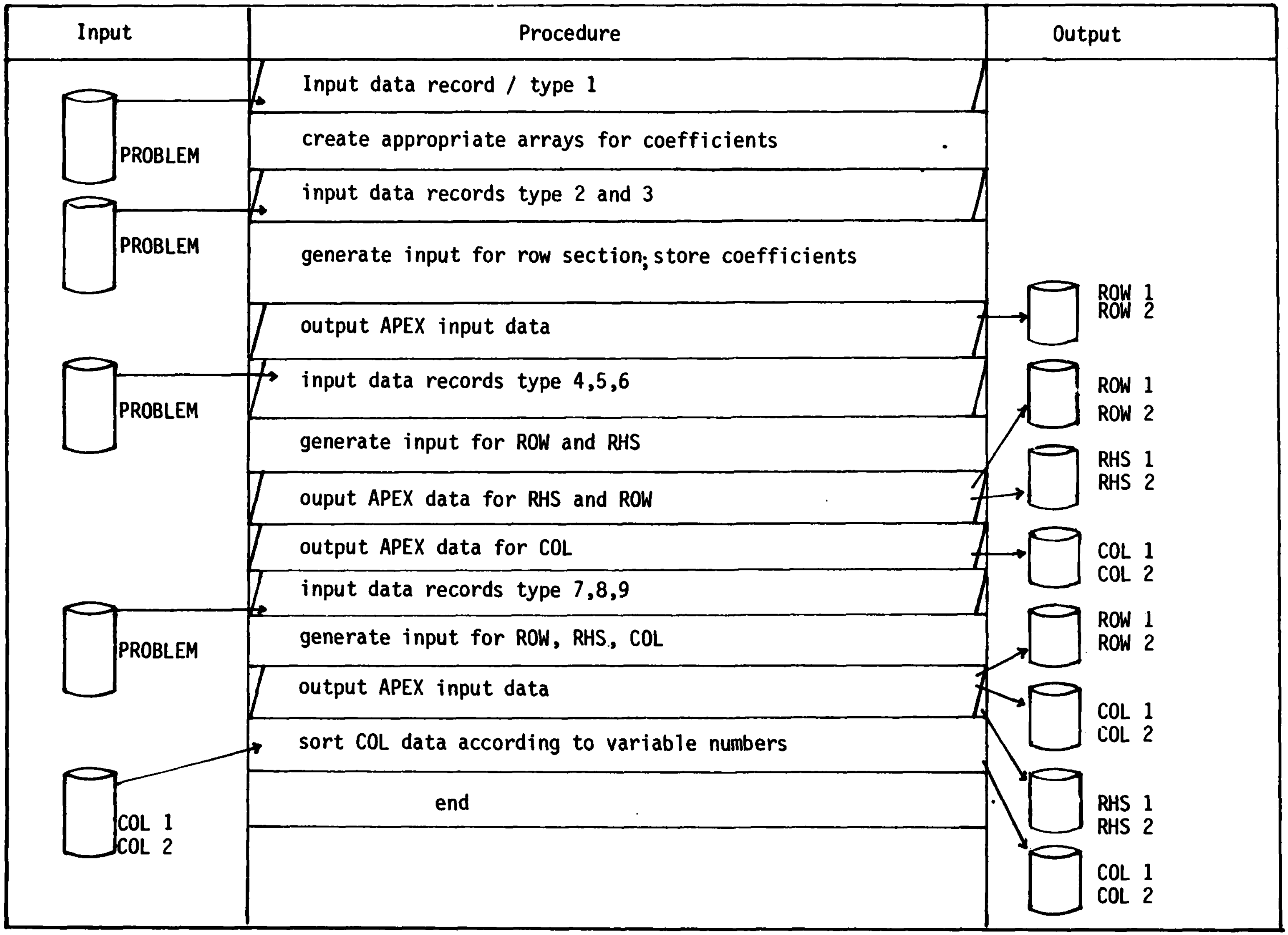
Language: PASCAL

Status:
- calling programs: MAINCCL
- called programs: --
  control procedure
  <u>main program</u>
  subroutine
- mode: Batch

Data files: PROBDAT, PROBLEM

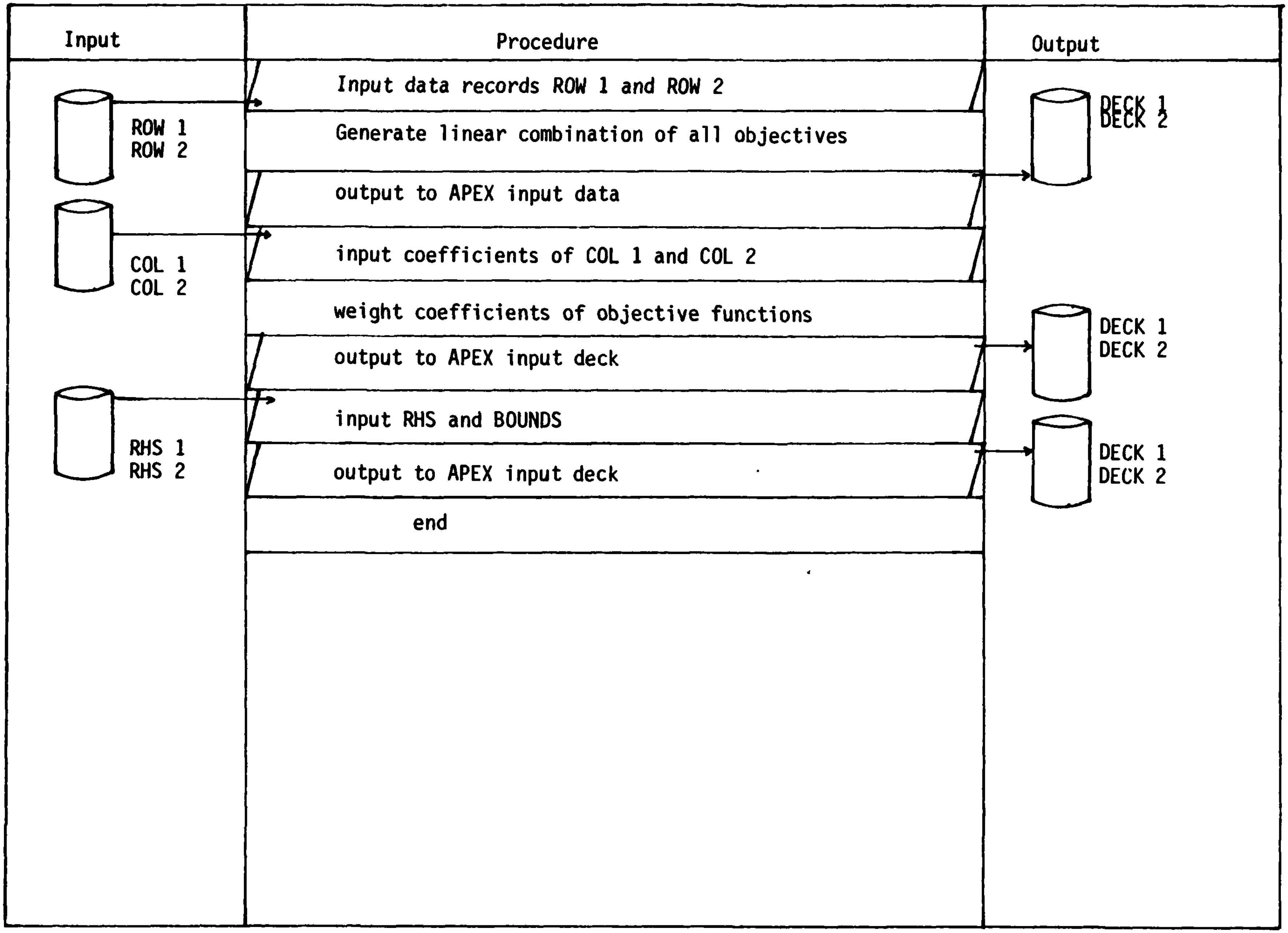
## IV.3.8 SINGLP

| | |
|---|---|
| Name: | SINGLP |
| Task: | create the individual LP's for determination of $\underline{C}$ and $\bar{C}$ |
| Language: | PASCAL |
| Status: | - calling programs: MAINCCL |
| | - called programs: -- |
| | control procedure |
| | <u>main program</u> |
| | subroutine |
| | - mode: Batch |
| Data files: | PROBLEM, ROW 1/ ROW 2, COL 1/COL 2, RHS 1/RHS 2 |


Input
Procedure
Output
PROBLEM
PROBLEM
PROBLEM
PROBLEM
COL 1
COL 2
Input data record / type 1
create appropriate arrays for coefficients
input data records type 2 and 3
generate input for row section; store coefficients
output APEX input data
input data records type 4,5,6
generate input for ROW and RHS
ouput APEX data for RHS and ROW
output APEX data for COL
input data records type 7,8,9
generate input for ROW, RHS, COL
output APEX input data
sort COL data according to variable numbers
end
ROW 1
ROW 2
ROW 1
ROW 2
RHS 1
RHS 2
COL 1
COL 2
ROW 1
ROW 2
COL 1
COL 2
RHS 1
RHS 2
COL 1
COL 2

## IV.3.9 INDVLP

| | |
|---|---|
| Name: | INDVLP |
| Task: | take over the weighted objective functions to the individual LP's |
| Language: | PASCAL |
| Status: | - calling programs: MAINCCL |
| | - called programs: -- |
| | control procedure |
| | main program |
| | subroutine |
| | - mode: Batch |
| Data files: | ROW 1/ROW 2, COL 1/COL 2, RHS 1/RHS 2, DECK 1/ DECK 2 |


Input
Procedure
Output
ROW 1
ROW 2
COL 1
COL 2
RHS 1
RHS 2
Input data records ROW 1 and ROW 2
Generate linear combination of all objectives
output to APEX input data
input coefficients of COL 1 and COL 2
weight coefficients of objective functions
output to APEX input deck
input RHS and BOUNDS
output to APEX input deck
end
DECK 1
DECK 2
DECK 1
DECK 2
DECK 1
DECK 2

## IV.3.10. COMPLP

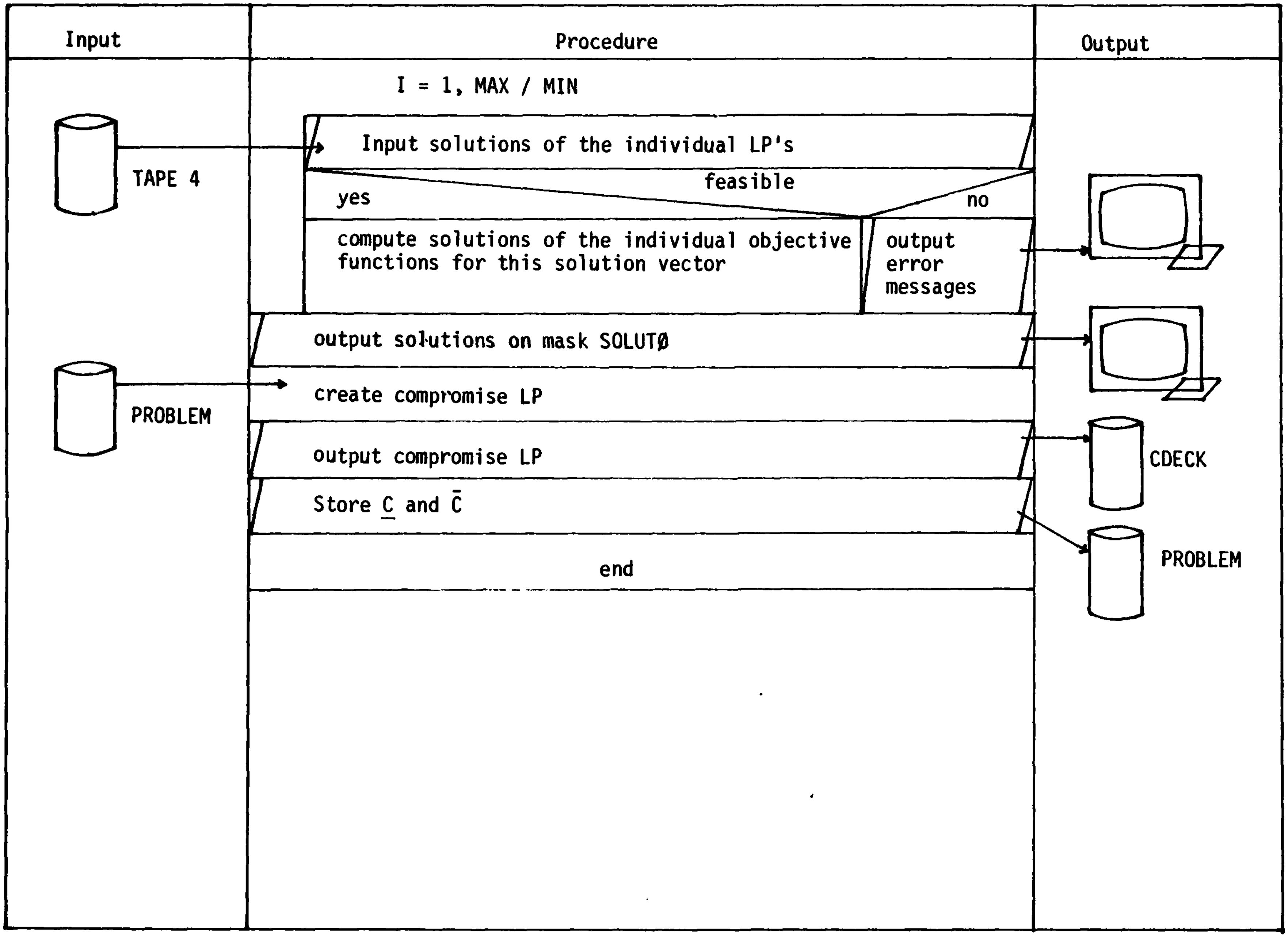
Name: COMPLP

Task: take over the results of the individual LP's; compute the coefficients of the compromise LP; create data structure of APEX

Language: FORTRAN

Status:
- calling programs: MAINCCL
- called programs: --
  - control procedure
  - main program
  - subroutine
- mode: Batch

Data files: TAPE 4, CDECK


Input
Procedure
Output
I = 1, MAX / MIN
TAPE 4
Input solutions of the individual LP's
feasible
yes
no
compute solutions of the individual objective functions for this solution vector
output error messages
output solutions on mask SOLUTØ
PROBLEM
create compromise LP
output compromise LP
CDECK
Store C and C̄
PROBLEM
end

## IV.3.11 SOLUT

Name: SOLUT

Task: take over the results of a compromise LP; processing and output of the solutions in dialog; modify the coefficients of the compromise LP's, if desired by the decision maker; create the modified data structure for APEX

Language: FORTRAN

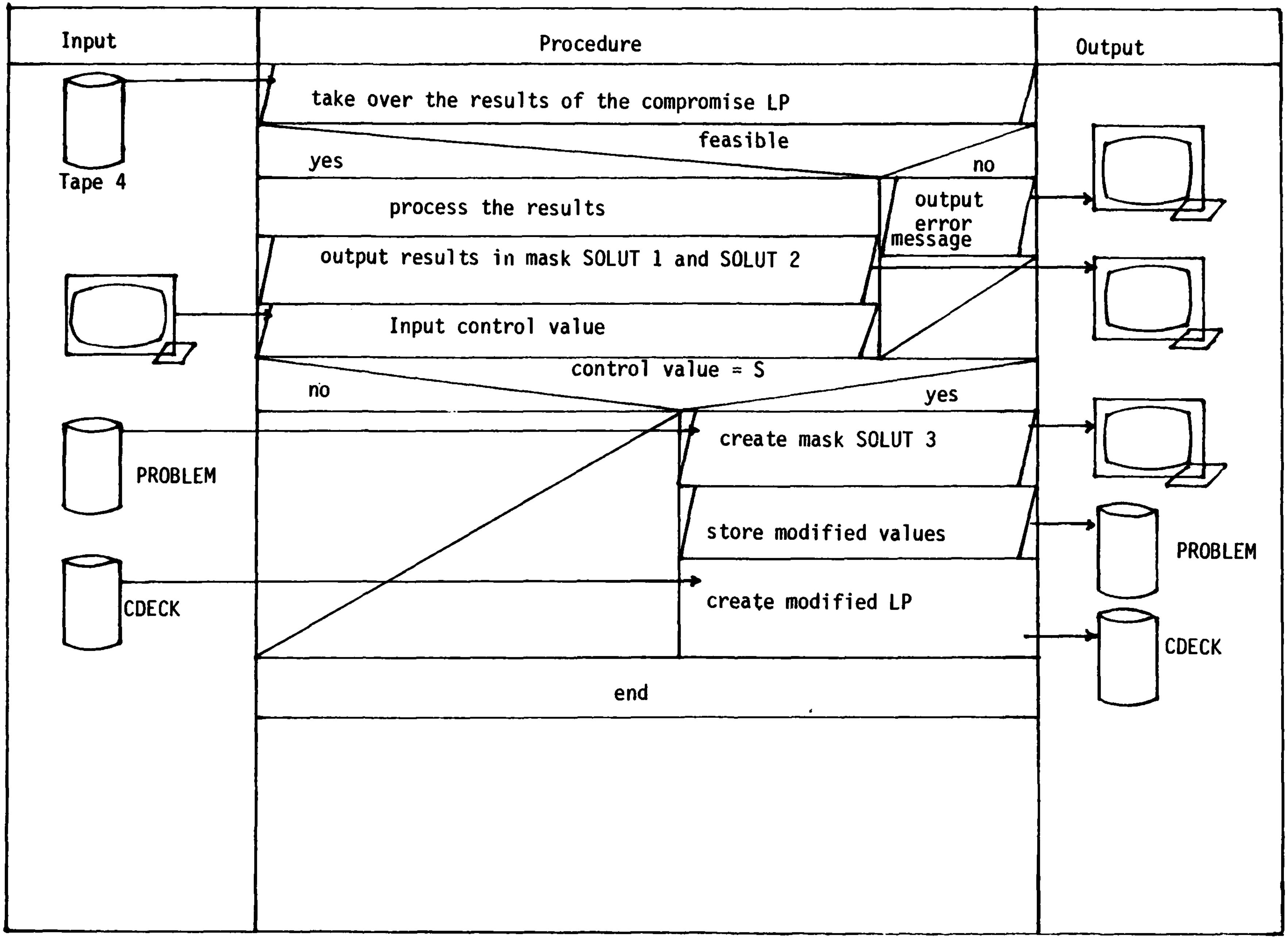
Status:
- calling programs: MAINCCL
- called programs: --

  control procedure

  <u>main program</u>

  subroutine
- mode: dialog

Data files: CDECK, TAPE 4, PROBLEM



Input
Procedure
Output
take over the results of the compromise LP
Tape 4
feasible
yes
no
process the results
output error message
output results in mask SOLUT 1 and SOLUT 2
Input control value
control value = S
no
yes
create mask SOLUT 3
PROBLEM
store modified values
PROBLEM
CDECK
create modified LP
CDECK
end

## IV.3.12 MAINCCL

Name: MAINCCL

Task: MAINCCL represents a local file, on which the job control commands still to be processed are stored. MAINCCL selects the individual programs and is changed by these programs according to the wishes of the decision maker. Examples of the contents of MAINCCL are given in the chapter on the structure of the system

Language: NOS 1.4 - CCL

Status:
- calling programs: ---
- called programs: MAINTEN 1, PROBSOL, SINGLP, INDVLP, COMPLP, SOLUT

  control procedure
  main program
  subroutine
- mode: Batch

Data files: ---

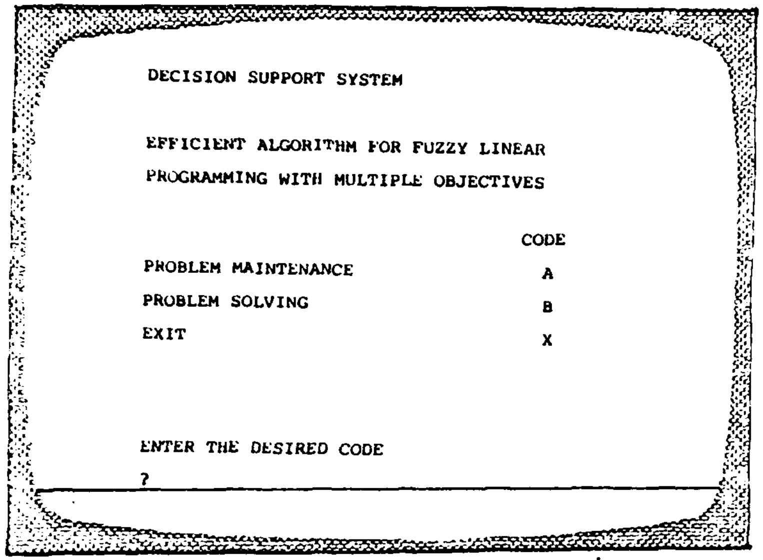
## IV.4 Error Messages

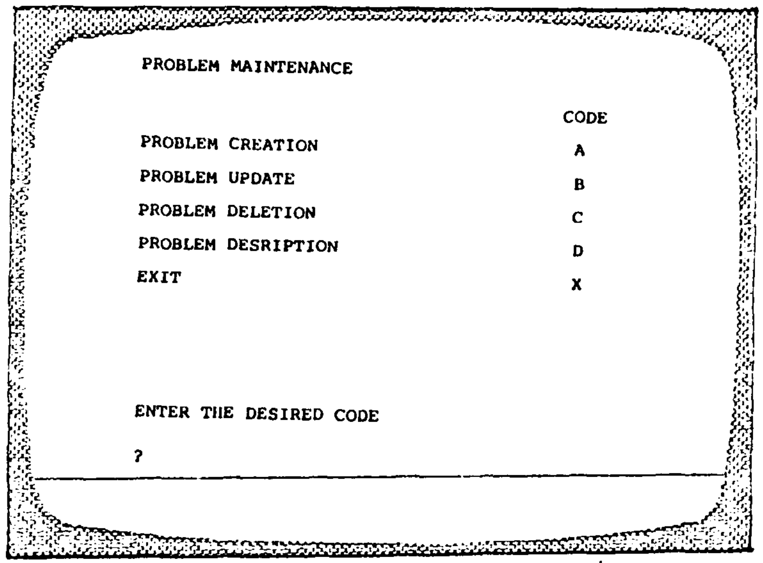
| Error message | Program interrupt | Expected activity of men | Origin |
|---|---|---|---|
| Please enter correct code (A,B,C,D,X) | no | revise input | MAINTEN 1 |
| Please enter numerical value | no | revise input | CREAT, UPDTE, COMPLP, SOLUT |
| The following signs are not allowed in the keyword ' ', ' ', ' ' | no | revise input | CREAT |
| Problem 'xxx' is not feasible | no | modify coefficients | COMPLP, SOLUT |
| Record type 'xxx' in PROBLEM missing | yes | revise file | SINGLP, SOLUT |
| Incorrect record type in PROBLEM | yes | revise file | SINGLP, SOLUT |
| Incorrect file structure 'xxx' | yes | revise file | SINGLP, INDVLP, COMPLP, SOLUT |

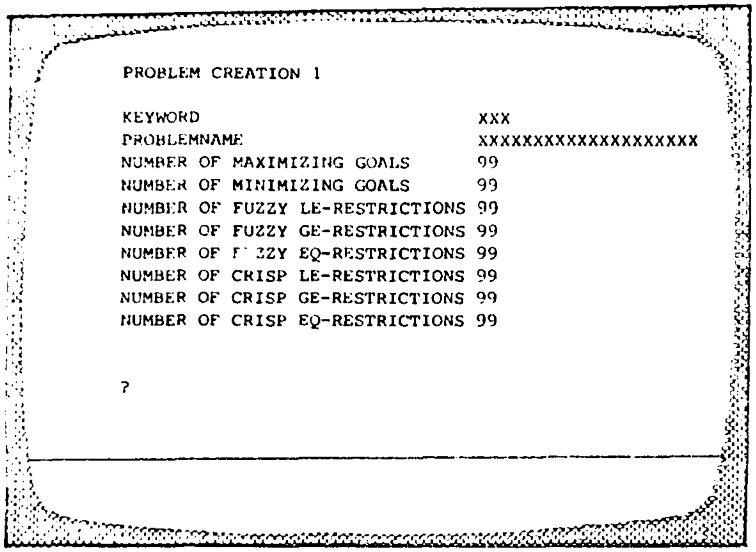
## IV.5 Masks for dialog

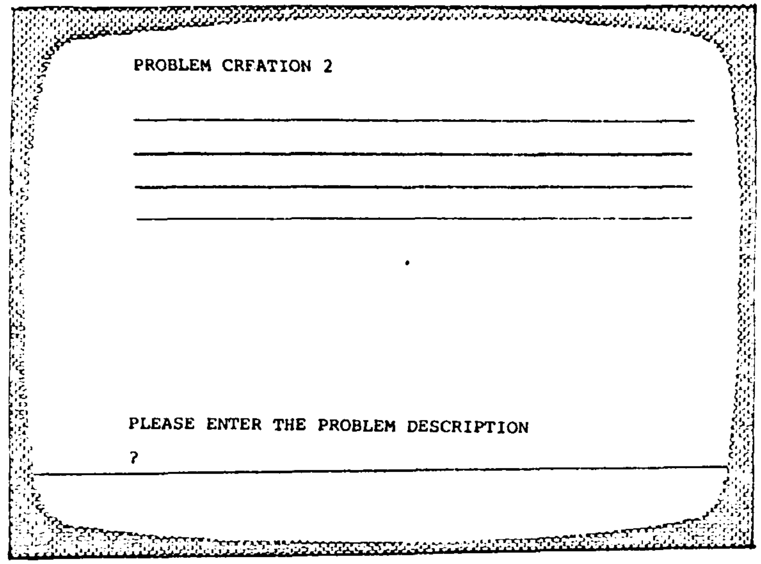
The input of data principally takes place in the last available row in all masks because of the difficulties described in the Sixth Periodic report, page 5. Normally this is row 21.
Rows 23 and 24 serve for the output of error messages and hints.

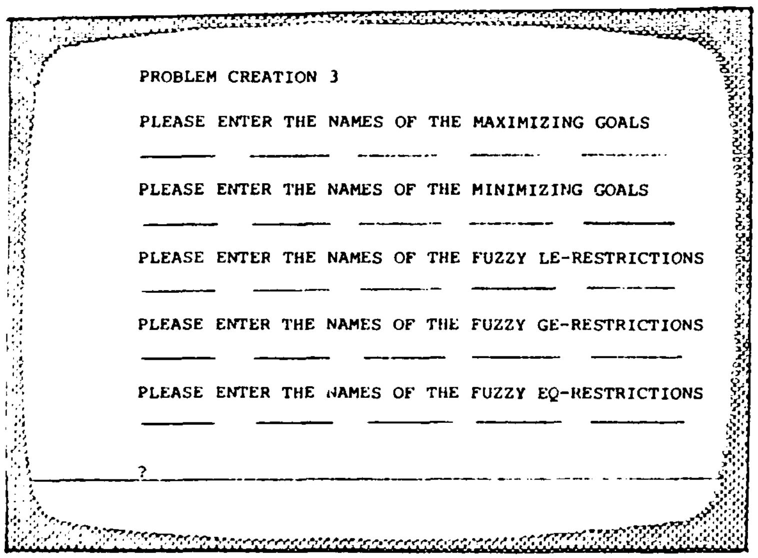
Remind that the inplemented masks are partially processed during the dialog. Thus in some instances the mask will be presented as created after some steps of dialog.

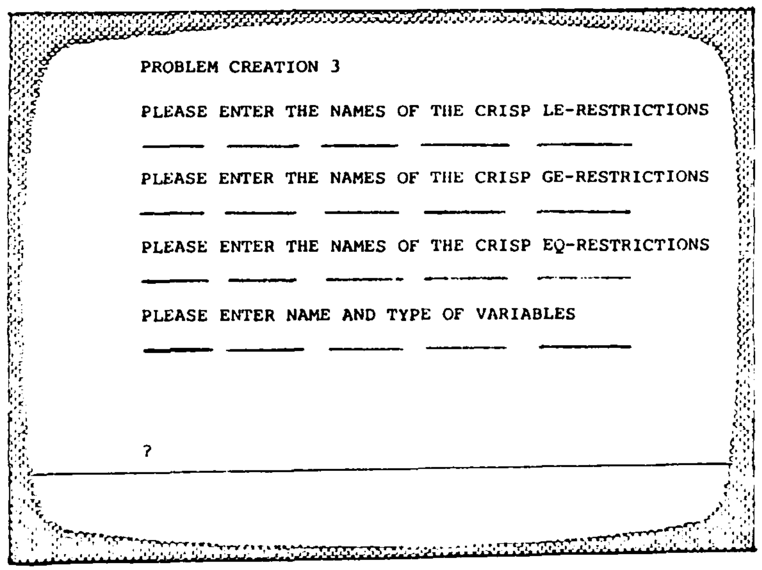

DECISION SUPPORT SYSTEM
EFFICIENT ALGORITHM FOR FUZZY LINEAR
PROGRAMMING WITH MULTIPLE OBJECTIVES
CODE
PROBLEM MAINTENANCE A
PROBLEM SOLVING B
EXIT X
ENTER THE DESIRED CODE
?

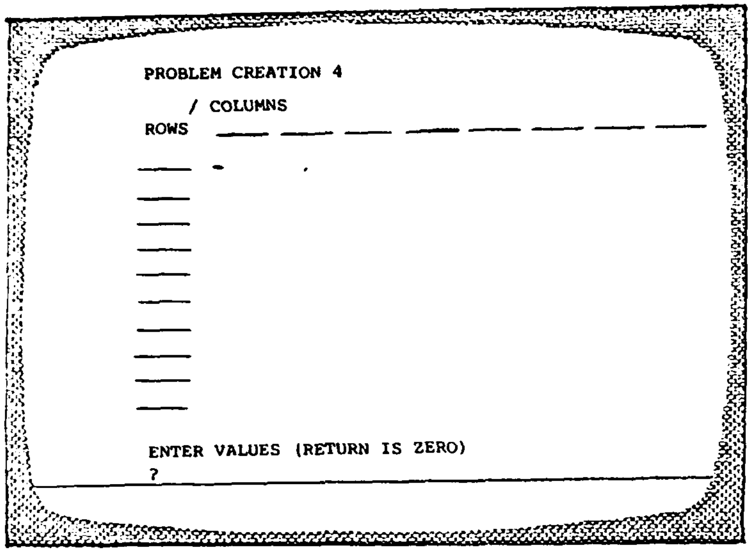

PROBLEM MAINTENANCE
CODE
PROBLEM CREATION A
PROBLEM UPDATE B
PROBLEM DELETION C
PROBLEM DESRIPTION D
EXIT X
ENTER THE DESIRED CODE
?

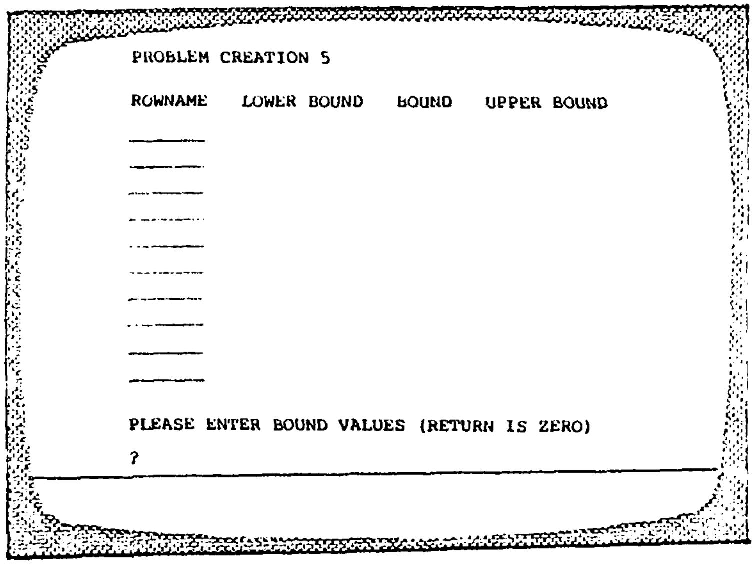

PROBLEM CREATION 1
KEYWORD XXX
PROBLEMNAME XXXXXXXXXXXXXXXXXXX
NUMBER OF MAXIMIZING GOALS 99
NUMBER OF MINIMIZING GOALS 99
NUMBER OF FUZZY LE-RESTRICTIONS 99
NUMBER OF FUZZY GE-RESTRICTIONS 99
NUMBER OF F ZZY EQ-RESTRICTIONS 99
NUMBER OF CRISP LE-RESTRICTIONS 99
NUMBER OF CRISP GE-RESTRICTIONS 99
NUMBER OF CRISP EQ-RESTRICTIONS 99
?

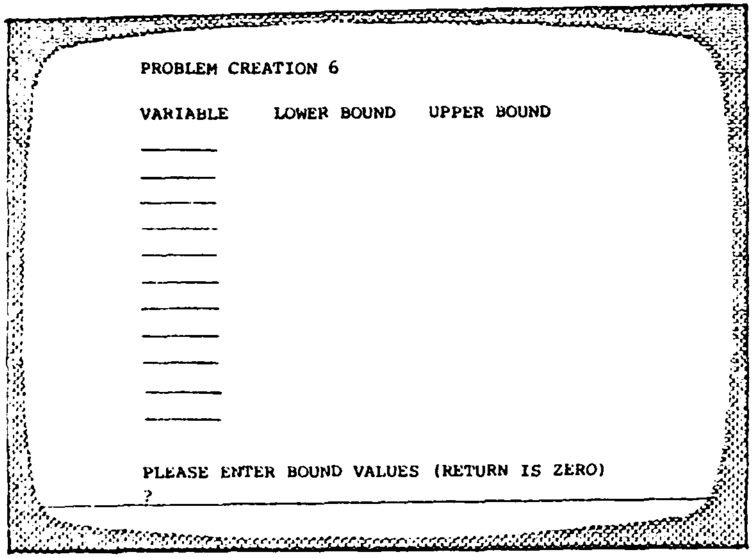

PROBLEM CRFATION 2
PLEASE ENTER THE PROBLEM DESCRIPTION
?


PROBLEM CREATION 3
PLEASE ENTER THE NAMES OF THE MAXIMIZING GOALS
PLEASE ENTER THE NAMES OF THE MINIMIZING GOALS
PLEASE ENTER THE NAMES OF THE FUZZY LE-RESTRICTIONS
PLEASE ENTER THE NAMES OF THE FUZZY GE-RESTRICTIONS
PLEASE ENTER THE NAMES OF THE FUZZY EQ-RESTRICTIONS
?


PROBLEM CREATION 3
PLEASE ENTER THE NAMES OF THE CRISP LE-RESTRICTIONS
PLEASE ENTER THE NAMES OF THE CRISP GE-RESTRICTIONS
PLEASE ENTER THE NAMES OF THE CRISP EQ-RESTRICTIONS
PLEASE ENTER NAME AND TYPE OF VARIABLES
?


PROBLEM CREATION 4
/ COLUMNS
ROWS
ENTER VALUES (RETURN IS ZERO)
?


PROBLEM CREATION 5
ROWNAME LOWER BOUND BOUND UPPER BOUND
PLEASE ENTER BOUND VALUES (RETURN IS ZERO)
?


PROBLEM CREATION 6
VARIABLE LOWER BOUND UPPER BOUND
PLEASE ENTER BOUND VALUES (RETURN IS ZERO)
?

PROBLEM UPDATE 1

| | CODE |
|---|---|
| CHANGE DESCRIPTION | A |
| CHANGE ROWS | B |
| CHANGE COLUMNS | C |
| CHANGE RHS | D |
| CHANGE COEFFICIENTS | E |
| EXIT | X |

ENTER THE DESIRED CODE
?

PROBLEM UPDATE 2

ENTER KEYWORD
______

PROBLEM DESCRIPTION

1 KEYWORD ___
2 PROBLEMNAME ____________
3 NUMBER OF MAXIMIZING GOALS ___
4 NUMBER OF MINIMIZING GOALS ___
5 NUMBER OF FUZZY LE-RESTRICTIONS ___
6 NUMBER OF FUZZY GE-RESTRICTIONS ___
7 NUMBER OF FUZZY EQ-RESTRICTIONS ___
8 NUMBER OF CRISP LE-RESTRICTIONS ___
9 NUMBER OF CRISP GE-RESTRICTIONS ___
Ø NUMBER OF CRISP EQ-RESTRICTIONS ___

ENTER NUMBER OF LINE TO CHANGE
?
ENTER VALUE
?

PROBLEM UPDATE 3

| ROWNAME | LOWER BOUND | BOUND | UPPER BOUND |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 1Ø | 11 | 12 |
| 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 2Ø |
| 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 |
| 29 | 3Ø | 31 | 32 |
| 33 | 34 | 35 | 36 |
| 37 | 38 | 39 | 4Ø |

ENTER THE NUMBER OF VALUE TO CHANGE/ ENTER VALUE
? ?

PROBLEM UPDATE 4

| VARIABLE | LOWER BOUND | UPPER BOUND |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
| 1Ø | 11 | 12 |
| 13 | 14 | 15 |
| 16 | 17 | 18 |
| 19 | 2Ø | 21 |
| 22 | 23 | 24 |
| 25 | 26 | 27 |
| 28 | 29 | 3Ø |

ENTER THE NUMBER OF VALUE TO CHANGE/ ENTEP VALUE
? ?

PROBLEM UPDATE 5

/ COLUMNS

| ROWS | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| ____ | 8 | 9 | 1Ø | 11 | 12 | 13 | 14 |
| ____ | 15 | 16 | 17 | 18 | 19 | 2Ø | 21 |
| ____ | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| ____ | 29 | 3Ø | 31 | 32 | 33 | 34 | 35 |
| ____ | 36 | 37 | 38 | 39 | 4Ø | 41 | 42 |
| ____ | 43 | 44 | 45 | 46 | 47 | 48 | 49 |
| ____ | 5Ø | 51 | 52 | 53 | 54 | 55 | 56 |
| ____ | 57 | 58 | 59 | 6Ø | 61 | 62 | 63 |
| ____ | 64 | 65 | 66 | 67 | 68 | 69 | 7Ø |

ENTER THE NUMBER OF VALUE TO CHANGE/ ENTER VALUE

? ?

PROBLEM DESCRIPTION 1

FOREWARD F BACKWARD B SEARCH S EXIT X

?

PROBLEM DESCRIPTION 2

ENTER KEYWORD

____

PROBLEM DESCRIPTION
KEYWORD ____
PROBLEMNAME ____________
NUMBER OF MAXIMIZING GOALS __
NUMBER OF MINIMIZING GOALS __
NUMBER OF FUZZY LE-RESTRICTIONS __
NUMBER OF FUZZY GE-RESTRICTIONS __
NUMBER OF FUZZY EQ-RESTRICTIONS __
NUMBER OF CRISP LE-RESTRICTIONS __
NUMBER OF CRISP GE-RESTRICTIONS __
NUMBER OF CRISP EQ-RESTRICTIONS __

PROBLEM DELETION

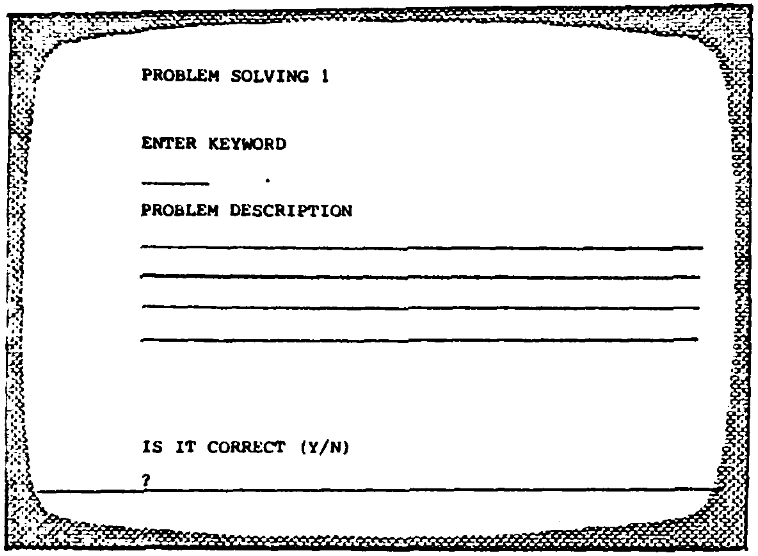
ENTER THE KEYWORD OF THE PROBLEM TO BE DELETED

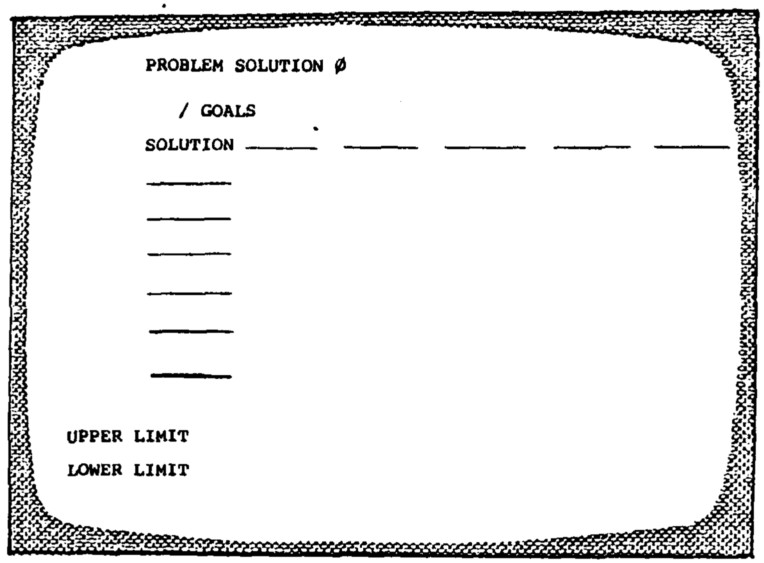
____

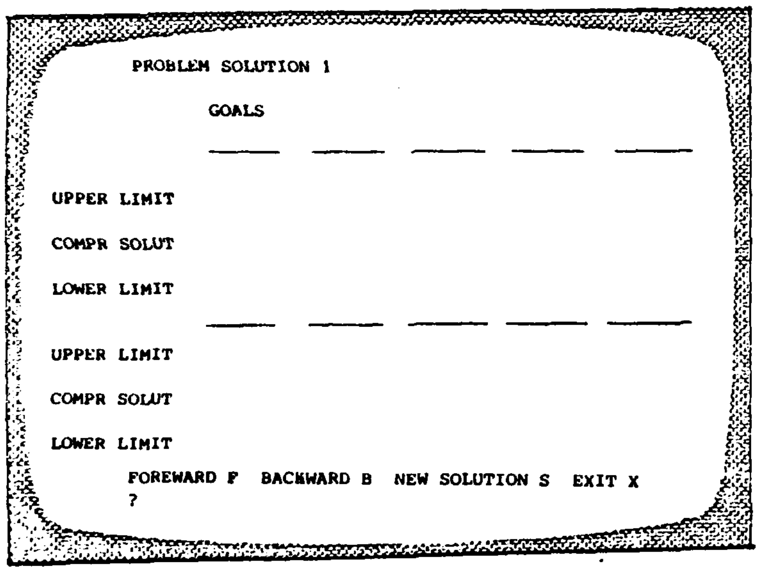
PROBLEMDESCRIPTION

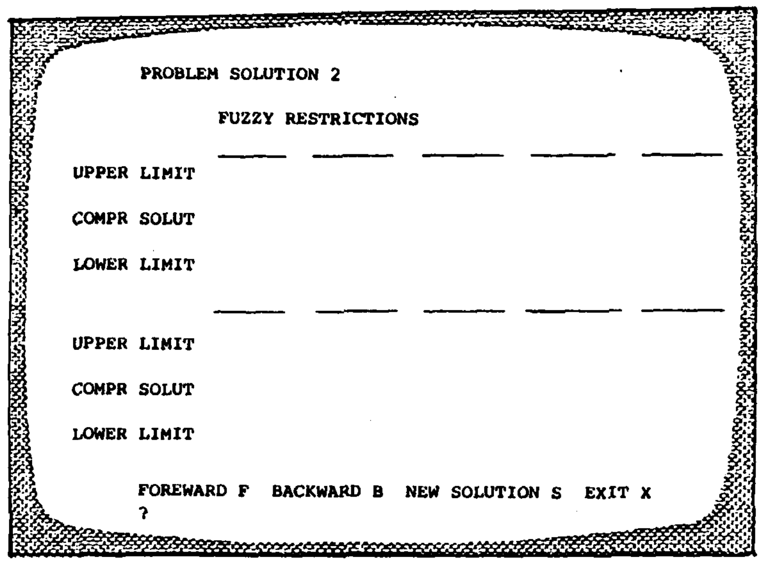
ARE YOU SHURE (Y/N)

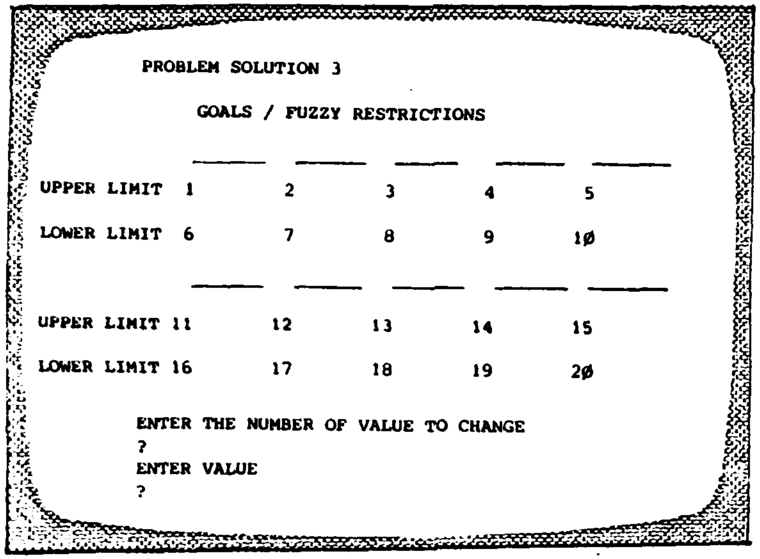
?


PROBLEM SOLVING 1
ENTER KEYWORD
PROBLEM DESCRIPTION
IS IT CORRECT (Y/N)
?


PROBLEM SOLUTION Ø
/ GOALS
SOLUTION
UPPER LIMIT
LOWER LIMIT


PROBLEM SOLUTION 1
GOALS
UPPER LIMIT
COMPR SOLUT
LOWER LIMIT
UPPER LIMIT
COMPR SOLUT
LOWER LIMIT
FOREWARD F BACKWARD B NEW SOLUTION S EXIT X
?


PROBLEM SOLUTION 2
FUZZY RESTRICTIONS
UPPER LIMIT
COMPR SOLUT
LOWER LIMIT
UPPER LIMIT
COMPR SOLUT
LOWER LIMIT
FOREWARD F BACKWARD B NEW SOLUTION S EXIT X
?


PROBLEM SOLUTION 3
GOALS / FUZZY RESTRICTIONS
UPPER LIMIT 1 2 3 4 5
LOWER LIMIT 6 7 8 9 1Ø
UPPER LIMIT 11 12 13 14 15
LOWER LIMIT 16 17 18 19 2Ø
ENTER THE NUMBER OF VALUE TO CHANGE
?
ENTER VALUE
?

## Appendix

Exemplarily the listing of 2 programs is shown in the appendix.

```
PROGRAM SINGLP(INPUT/,OUTPUT,DATEN1,DATEN2);
CONST BLANK10='          ';
      N=1000;
      EPSI=1.0E-04;
      ROWTYP1='GOALMAX';
      ROWTYP2='GOALMIN';
      ROWTYP3='FRESTLE';
      ROWTYP4='FRESTGE';
      ROWTYP5='FRESTEQ';
      ROWTYP6='CRESTLE';
      ROWTYP7='CRESTGE';
      ROWTYP8='CRESTEQ';
      COLTYP1='VARIABLE';
      RHSTYP1='RHSFRES';
      RHSTYP2='RHSCRES';
TYPE DEFCOEFCHAR=RECORD SPALTIND:INTEGER;
                        COEFWERT:REAL        END (*OF COLUMN GEFUNDEN*);
     DEFCOEFCHF=ARRAY[1..N] OF DEFCOEFCHAR;
     DEFSATZTYP = 1..9;
     CHARAR=PACKED ARRAY[1..7] OF CHAR;
     CHARFI=FILE OF CHAR;
     DATSTRUK1=RECORD
               SATZTYP:DEFSATZTYP;
               SATZLAENGE:INTEGER;
               CASE STYP:DEFSATZTYP OF
               1:(MAXFZ,MINFZ,FRLE,FRGE,FREQ,CRLE,CRGE,CREQ:INTEGER);
               2:(MAXCOEF:DEFCOEFCHF);
               3:(MINCOEF:DEFCOEFCHF);
               4:(FRLEBOU,FRLEBOQ:REAL;FRLECOEF:DEFCOEFCHF);
               5:(FRGEBOU,FRGEBOQ:REAL;FRGECOEF:DEFCOEFCHF);
               6:(FREQBOU,FREQB,FREQBUU:REAL;FREQCOEF:DEFCOEFCHF);
               7:(CRLEB:REAL;CRLECOEF:DEFCOEFCHF);
               8:(CRGEB:REAL;CRGECOEF:DEFCOEFCHF);
               9:(CREQB:REAL;CREQCOEF:DEFCOEFCHF)          END;
     DATSATZ=FILE OF DATSTRUK1;
VAR IMAXFZ,IMINFZ,IFRLE,IFRGE,IFREQ,ICRLE,ICRGE,ICREQ,LGE,I,J,ERRCODE,RECCODE,
    RECLGE,K:INTEGER;
    ROWW,PRUEF3:CHARAR;
    ZWI,ZWI2:PACKED ARRAY[1..5] OF CHAR;
    ZWI3:PACKED ARRAY[1..2] OF CHAR;
    PRUEF1,PRUEF2:CHAR;
    SCHL,KK,SCHLV,LFNR,TYP:INTEGER;
    WERT,MULTI:REAL;
    NAME:PACKED ARRAY[1..40] OF CHAR;
VAR DATEN1:CHARFI;
    DATEN2:CHARFI;
```

```
BEGIN (*OF MAIN PR.*)
(* EINLESEN DER STEUERUNGSPARAMETER *)
RESET(DATEN1);
READLN;READ(K,LFNR,TYP);
(* BERECHNUNG STEUERUNGSWERTE *)
   MULTI:=1-(K-1)*EPSI;
   SCHL:=ABS(LFNR-333*TYP);
   IF TYP=2
      THEN RUWW:=RUWTYP1
      ELSE RUWW:=RUWTYP2;
(* VERARBEITUNG DATENSAETZE *)
   WHILE NOT EOF(DATEN1)DO
         BEGIN (*OF READHER*)
         FOR I:=1 TO 40 DO NAME[I]:=' ';
         READ(DATEN1,PRUEF1,PRUEF2);
         (* ABFRAGE,OB ZIELFKT.COEF*)
         IF(PRUEF1 <> '*')OR((PRUEF1='*')AND((PRUEF2='X')OR(PRUEF2='D')))
            THEN BEGIN (* OF UEBERNAHME*)
            I:=1;
            IF (PRUEF1='*') THEN PRUEF1:=' ';
            IF (PRUEF2='X') THEN PRUEF2:=' ';
            WHILE NOT EOLN(DATEN1)DO
                  BEGIN (*OF ZEICHENEINL*)
                  READ(DATEN1,NAME[I]);
                  I:=I+1;
                  END (*OF ZEICHENEINLES*);
                  READLN(DATEN1);
            WRITE(DATEN2,PRUEF1,PRUEF2);
            FOR J:=1 TO (I-1) DO WRITE(DATEN2,NAME[J]);
            WRITELN(DATEN2,NAME[I]);
            END(*OF UEBERNAHME*)
            ELSE BEGIN (*OF UEBERPRUEFUNG*)
            (* ABFRAGE, OB ROW SECTION *)
               IF(PRUEF2='N')
                  THEN BEGIN (*OF ZIEL GEFUNDEN*)
                        READ(DATEN1,ZWI3[1],ZWI3[2]);
                        FOR KK := 1 TO 7 DO READ(DATEN1,PRUEF3[KK]);
                        READLN(DATEN1,SCHLV);
                        PRUEF1:=' ';
                        WRITELN(DATEN2,PRUEF1,PRUEF2,ZWI3:2,PRUEF3:7,SCHLV:3)
                        END (*OF ZIEL GEFUNDEN*)
                  ELSE BEGIN (*OF COLUMN GEFUNDEN*)
                        FOR KK:=1 TO 6 DO READ(DATEN1,ZWI[KK]);
                        FOR KK:=1 TO 6 DO READ(DATEN1,ZWI2[KK]);
                        FOR KK:=1 TO 7 DO READ(DATEN1,PRUEF3[KK]);
                        READLN(DATEN1,SCHLV,WERT);
                        IF (PRUEF3=RUWW)AND(SCHLV=SCHL)
                           THEN BEGIN (*OF HAUPTZIEL*)
                                WERT:=WERT*MULTI;
                                PRUEF1:=' ';
                                END (*OF HAUPTZIEL*)
                           ELSE BEGIN (*OF NEBENZIEL*)
                                WERT:=WERT*EPSI;
                                PRUEF1:=' ';
                                END (*OF NEBENZIEL*);
                  WRITELN(DATEN2,PRUEF1,PRUEF2,ZWI:6,ZWI2:6,PRUEF3:7,SCHLV:3
                 ,WERT:12:4);
                  END;
            END(* OF UEBERPRUEFUNG *);
      END (*OF READHER*)
   END.
```

```
PROGRAM INDVLP(DATENINPUT,OUTPUT,DATENROW1,DATENROW2,DATENCOL1,DATENCOL2,
               DATENRHS1,DATENRHS2);
CONST BLANK10=≡          ≡;
      N=1000;
      ROWTYP1=≡GOALMAX≡;
      ROWTYP2=≡GOALMIN≡;
      ROWTYP3=≡FRESTLE≡;
      ROWTYP4=≡FRESTGE≡;
      ROWTYP5=≡FRESTEQ≡;
      ROWTYP6=≡CRESTLE≡;
      ROWTYP7=≡CRESTGE≡;
      ROWTYP8=≡CRESTEQ≡;
      COLTYP1=≡VARIABL≡;
      RHSTYP1=≡RHSFRES≡;
TYPE DEFCOEFCHAR=RECORD SPALTIND:INTEGER;
                        COEFWERT:REAL      END;
     DEFCOEFCHF=ARRAY[1..N] OF DEFCOEFCHAR;
     DEFSATZTYP = 1..9;
     CHARAR=PACKED ARRAY[1..7] OF CHAR;
     CHARFI=FILE OF CHAR;
     DATSTRUK1=RECORD
               SATZTYP:DEFSATZTYP;
               SATZLAENGE:INTEGER;
               CASE STYP:DEFSATZTYP OF
               1:(MAXFZ,MINFZ,FRLE,FRGE,FREQ,CRLE,CRGE,CREQ:INTEGER);
               2:(MAXCOEF:DEFCOEFCHF);
               3:(MINCOEF:DEFCOEFCHF);
               4:(FRLEBUQ,FRLEBOQ:REAL;FRLECOEF:DEFCOEFCHF);
               5:(FRGEBUQ,FRGEBOQ:REAL;FRGECOEF:DEFCOEFCHF);
               6:(FREQBUQ,FREQB,FREQBOQ:REAL;FREQCOEF:DEFCOEFCHF);
               7:(CRLEB:REAL;CRLECOEF:DEFCOEFCHF);
               8:(CRGEB:REAL;CRGECOEF:DEFCOEFCHF);
               9:(CREQB:REAL;CREQCOEF:DEFCOEFCHF)          END;
     DATSATZ=FILE OF DATSTRUK1;
VAR IMAXFZ,IMINFZ,IFRLE,IFRGE,IFREQ,ICRLE,ICRGE,ICREQ,LGE,I,J,ERRCODE,RECCODE,
    RECLGE,MJE,K:INTEGER;
VAR DATENINPUT:DATSATZ;
    DATENROW1:CHARFI;
    DATENROW2:CHARFI;
    DATENCOL1:CHARFI;
    DATENCOL2:CHARFI;
    DATENRHS1:CHARFI;
    DATENRHS2:CHARFI;
PROCEDURE DATLES(VAR DATREC:DATSATZ;VAR RECTYP,SATZLGE,ERRTYP:INTEGER);
(* EINLESEN EINES BELIEBIGEN SATZES DER PROBLEMDATEI *)
BEGIN
    GET(DATREC);
    (* UEBERPRUEFUNG, OB DATEI LEER *)
    IF EOF(DATREC)
         THEN BEGIN
              WRITELN(≡***ERROR 4 :  RECORD TYPE ≡,RECTYP:1,≡ MISSING   ****≡);
              ERRTYP := 1
              END
         (* UEBERPRUEFUNG, OB SATZTYP KORREKT*)
         ELSE IF RECTYP <> DATREC↑.SATZTYP
              THEN BEGIN
                   WRITELN(≡***ERROR 5 :  INCORRECT RECORD TYPE   ****≡);
                   ERRTYP := 2
                   END
              ELSE SATZLGE := DATREC↑.SATZLAENGE;
         END (* OF PROCEDURE DATLES *);
PROCEDURE ZIELFKT(VAR DATREC:DATSATZ;VAR ROW1,ROW2,COL1,COL2:CHARFI;
                  VAR LGE,LFNR,TYP:INTEGER);
VAR ZIEL,J,SCHL:INTEGER;WERT:REAL;ROWW:CHARAR;
(* VERARBEITUNG DER LFNR-TEN ZIELFUNKTION *)
(* SCHRITT1 : EINTRAG IN ROW DATEIEN*)
(* UEBERPRUEFUNG, OB MAX-ZIELFUNKTION *)
```

```
BEGIN
    ZIEL := ABS(LFNR - 333 * TYP);
    IF TYP = 2
            THEN ROWW := ROWTYP1
            ELSE ROWW := ROWTYP2;
            WRITELN(ROW1,≡*N  ≡,ROWW:7,ZIEL:3);
            WRITELN(ROW2,≡*N  ≡,ROWW:7,ZIEL:3);
(* SCHRITT2 : EINTRAG IN COL DATEIEN*)
    FOR J := 1 TO LGE DO
            BEGIN
              SCHL := ABS(DATREC↑.MAXCOEF[J].SPALTIND - 999);
              IF TYP = 3
                    THEN WERT := -DATREC↑.MAXCOEF[J].COEFWERT
                    ELSE WERT := DATREC↑.MAXCOEF[J].COEFWERT;
              WRITELN(COL1,≡*   ≡,COLTYP1:7,SCHL:3,ROWW:7,ZIEL:3,WERT:12:4);
              WRITELN(COL2,≡*   ≡,COLTYP1:7,SCHL:3,ROWW:7,ZIEL:3,WERT:12:4);
            END;
END (* OF PROCEDURE ZIELFKT *);
PROCEDURE FREST(VAR DATREC:DATSATZ;VAR ROW1,ROW2,COL1,COL2,RHS1,RHS2:
                CHARFI;LGE,LFNR,TYP:INTEGER);
VAR ZIEL,HILF,J,SCHL:INTEGER;WERT:REAL;ROWW:CHARAR;
(* VERARBEITUNG DER LFNR-TEN FUZZY RESTRIKTION*)
(* SCHRITT1 : EINTRAG IN ROW DATEIEN*)
BEGIN
    ZIEL := ABS(LFNR - 999);
    HILF:=111;
    CASE TYP OF 4: BEGIN
                   WRITELN(ROW1,≡ L  ≡,ROWTYP3:7,ZIEL:3);
                   WRITELN(ROW2,≡ L  ≡,ROWTYP3:7,ZIEL:3);
                   ROWW := ROWTYP3;
                   END;
                5: BEGIN
                   WRITELN(ROW1,≡ G  ≡,ROWTYP4:7,ZIEL:3);
                   WRITELN(ROW2,≡ G  ≡,ROWTYP4:7,ZIEL:3);
                   ROWW := ROWTYP4  .
                   END;
                6: BEGIN
                   WRITELN(ROW1,≡ E  ≡,ROWTYP5:7,ZIEL:3);
                   WRITELN(ROW2,≡ L  ≡,ROWTYP3:6,≡L≡,ZIEL:3);
                   WRITELN(ROW2,≡ G  ≡,ROWTYP4:6,≡G≡,ZIEL:3)
                   END;
    END;
(* SCHRITT2 : EINTRAG IN RHS DATEIEN*)
    CASE TYP OF 4: BEGIN
                   WRITELN(RHS1,≡     ≡,RHSTYP1:7,HILF:3,ROWTYP3:7,ZIEL:3,
                          DATREC↑.FRLEBUQ:12:4);
                   WRITELN(RHS2,≡     ≡,RHSTYP1:7,HILF:3,ROWTYP3:7,ZIEL:3,
                          DATREC↑.FRLEBOQ:12:4);
                   END;
                5: BEGIN
                   WRITELN(RHS1,≡     ≡,RHSTYP1:7,HILF:3,ROWTYP4:7,ZIEL:3,
                          DATREC↑.FRLEBOQ:12:4);
                   WRITELN(RHS2,≡     ≡,RHSTYP1:7,HILF:3,ROWTYP4:7,ZIEL:3,
                          DATREC↑.FRLEBUQ:12:4);
                   END;
                6: BEGIN
                   WRITELN(RHS1,≡     ≡,RHSTYP1:7,HILF:3,ROWTYP5:7,ZIEL:3,
                          DATREC↑.FREQB:12:4);
                   WRITELN(RHS2,≡     ≡,RHSTYP1:7,HILF:3,ROWTYP3:6,≡L≡,
                          ZIEL:3,DATREC↑.FREQBOQ:12:4);
                   WRITELN(RHS2,≡     ≡,RHSTYP1:7,HILF:3,ROWTYP4:6,≡G≡,
                          ZIEL:3,DATREC↑.FREQBUQ:12:4)
                   END;
    END;
(* SCHRITT3 : EINTRAG IN COL DATEIEN*)
    FOR J := 1 TO LGE DO
            BEGIN
```

```
                    IF TYP = 6
                    THEN BEGIN
                          SCHL := ABS(DATREC↑.FREQCOEF[J].SPALTIND -   ,9);
                          WERT := DATREC↑.FREQCOEF[J].COEFWERT
                          END              ELSE BEGIN
                         SCHL := ABS(DATREC↑.FRLECOEF[J].SPALTIND - 999);
                         WERT := DATREC↑.FRLECOEF[J].COEFWERT
                        END;
     CASE TYP OF 4: BEGIN
                     WRITELN(COL1,≡      ≡,COLTYP1:7,SCHL:3,ROWW:7,ZIEL:3,WERT:12:4);
                     WRITELN(COL2,≡      ≡,COLTYP1:7,SCHL:3,ROWW:7,ZIEL:3,WERT:12:4)
                       END;
                   5: BEGIN
                     WRITELN(COL1,≡      ≡,COLTYP1:7,SCHL:3,ROWW:7,ZIEL:3,WERT:12:4);
                     WRITELN(COL2,≡      ≡,COLTYP1:7,SCHL:3,ROWW:7,ZIEL:3,WERT:12:4)
                       END;
                   6: BEGIN
          WRITELN(COL1,≡      ≡,COLTYP1:7,SCHL:3,ROWTYP5:7,ZIEL:3,WERT:12:4);
          WRITELN(COL2,≡      ≡,COLTYP1:7,SCHL:3,ROWTYP3:6,≡L≡,ZIEL:3,WERT:12:4);
          WRITELN(COL2,≡      ≡,COLTYP1:7,SCHL:3,ROWTYP4:6,≡G≡,ZIEL:3,WERT:12:4);
                       END;
        END;
     END;
END (* OF PROCEDURE FREST *);
PROCEDURE CREST(VAR DATREC:DATSATZ;VAR ROW1,ROW2,COL1,COL2,RHS1,RHS2:
                CHARFI;VAR LGE,LFNR,TYP:INTEGER);
VAR ZIEL,HILF,J,SCHL:INTEGER;WERT:REAL;ROWW:CHARAR;
(* VERARBEITUNG DER LFNR-TEN CRISP RESTRIKTION*)
(* SCHRITT1 : EINTRAG IN ROW DATEIEN*)
BEGIN
     ZIEL := ABS(LFNR - 999);
     HILF:=111;
     CASE TYP OF 7: BEGIN
                     WRITELN(ROW1,≡ L  ≡,ROWTYP6:7,ZIEL:3);
                     WRITELN(ROW2,≡ L  ≡,ROWTYP6:7,ZIEL:3);
                     ROWW := ROWTYP6;
                     END;
                  8: BEGIN
                     WRITELN(ROW1,≡ G  ≡,ROWTYP7:7,ZIEL:3);
                     WRITELN(ROW2,≡ G  ≡,ROWTYP7:7,ZIEL:3);
                     ROWW := ROWTYP7
                     END;
                  9: BEGIN
                     WRITELN(ROW1,≡ E  ≡,ROWTYP8:7,ZIEL:3);
                     WRITELN(ROW2,≡ E  ≡,ROWTYP8:7,ZIEL:3);
                  ROWW := ROWTYP8
                     END;
     END;
(* SCHRITT2 : EINTRAG IN RHS DATEIEN*)
     CASE TYP OF 7: BEGIN
                     WRITELN(RHS1,≡      ≡,RHSTYP1:7,HILF:3,ROWTYP6:7,ZIEL:3,
                             DATREC↑.CRLEB:12:4);
                     WRITELN(RHS2,≡      ≡,RHSTYP1:7,HILF:3,ROWTYP6:7,ZIEL:3,
                             DATREC↑.CRLEB:12:4);
                     END;
                  8: BEGIN
                     WRITELN(RHS1,≡      ≡,RHSTYP1:7,HILF:3,ROWTYP7:7,ZIEL:3,
                             DATREC↑.CRGEB:12:4);
                     WRITELN(RHS2,≡      ≡,RHSTYP1:7,HILF:3,ROWTYP7:7,ZIEL:3,
                             DATREC↑.CRGEB:12:4);
                     END;
                  9: BEGIN
                     WRITELN(RHS1,≡      ≡,RHSTYP1:7,HILF:3,ROWTYP8:7,ZIEL:3,
                             DATREC↑.CREQB:12:4);
                     WRITELN(RHS2,≡      ≡,RHSTYP1:7,HILF:3,ROWTYP8:7,
                             ZIEL:3,DATREC↑.CREQB:12:4);
                     END;
```

```
        END;
(* SCHRITT3 : EINTRAG IN COL DATEIEN*)
    FOR J := 1 TO LGE DO
            BEGIN
                  SCHL := ABS(DATRECT^.CRLECOEF[J].SPALTIND - 999);
                  WERT := DATRECT^.CRLECOEF[J].COEFWERT;
    CASE TYP OF 7: BEGIN
                WRITELN(COL1,'       ',COLTYP1:7,SCHL:3,ROWW:7,ZIEL:3,WERT:12:4);
                WRITELN(COL2,'       ',COLTYP1:7,SCHL:3,ROWW:7,ZIEL:3,WERT:12:4)
                  END;
               8: BEGIN
                WRITELN(COL1,'       ',COLTYP1:7,SCHL:3,ROWW:7,ZIEL:3,WERT:12:4);
                WRITELN(COL2,'       ',COLTYP1:7,SCHL:3,ROWW:7,ZIEL:3,WERT:12:4)
                  END;
               9: BEGIN
        WRITELN(COL1,'       ',COLTYP1:7,SCHL:3,ROWW:7,ZIEL:3,WERT:12:4);
        WRITELN(COL2,'       ',COLTYP1:7,SCHL:3,ROWW:7,ZIEL:3,WERT:12:4);
                  END;
     END;
    END;
END (* OF PROCEDURE CREST *);
BEGIN
(* VORBESETZUNG VON VARIABLEN*)
ERRCODE := 0;
(* EINLESEN DES ERSTEN DATENSATZES DER PROBLEMDATEI *)
RESET(DATENINPUT);
(* UEBERPRUEFUNG, OB DATEI LEER *)
IF EOF(DATENINPUT)
   THEN WRITELN('***ERROR 1 :  NO SOURCE DECK    ***')
   ELSE BEGIN
   (* UEBERPRUEFUNG, OB SATZTYP 1 VORHANDEN *)
   IF DATENINPUT^.SATZTYP <> 1
      THEN WRITELN ('***ERROR 2 :  INCORRECT FILE STRUCTURE    ***')
      ELSE BEGIN
      (* UEBERPRUEFUNG, OB SATZTYP 1 KORREKT *)
      IF DATENINPUT^.SATZLAENGE <> 10
        THEN WRITELN ('***ERROR 3 :  INCORRECT RECORD ',
         'STRUCTURE    ***')
         ELSE BEGIN
         (* UEBERNAHME PROBLEMCHARACTERISIERUNG *)
           IMAXFZ:=DATENINPUT^.MAXFZ;
           IMINFZ:=DATENINPUT^.MINFZ;
           IFRLE:=DATENINPUT^.FRLE;
           IFRGE:=DATENINPUT^.FRGE;
           IFREQ:=DATENINPUT^.FREQ;
           ICRLE:=DATENINPUT^.CRLE;
           ICRGE:=DATENINPUT^.CRGE;
           ICREQ:=DATENINPUT^.CREQ;
           (*INITIALISIERUNG DER ROW DATEIEN*)
           WRITELN(DATENROW1,'NAME',BLANK10:10,'MINRESTRIK');
           WRITELN(DATENROW1,'ROWS');
           WRITELN(DATENROW2,'NAME',BLANK10:10,'MAXRESTRIK');
           WRITELN(DATENROW2,'ROWS');
           (*ERZEUGUNG DER LINEARKOMBINATION*)
           WRITE(DATENROW1,'*ON ');
           WRITE(DATENROW2,'*ON ');
           FOR I := 1 TO IMAXFZ DO BEGIN
               J:=ABS(I-333*2);
               WRITE(DATENROW1,'GOALKOMBIN',ROWTYP1:7,J:3,'1.0');
               WRITELN(DATENROW1);
               WRITE(DATENROW1,'*X  ');
               WRITE(DATENROW2,'GOALKOMBIN',ROWTYP1:7,J:3,'1.0');
               WRITELN(DATENROW2);
               WRITE(DATENROW2,'*X  ');
               END;
           FOR I := 1 TO IMINFZ DO BEGIN
               J:=ABS(I-333*3);
```

```
        WRITE(DATENROW1,≡GOALKOMBINE≡,ROWTYP2:7,J:3,≡1.0≡);
        WRITELN(DATENROW1);
        WRITE(DATENROW1,≡*X   ≡);
        WRITE(DATENROW2,≡GOALKOMBINE≡,ROWTYP2:7,J:3,≡1.0≡);
        WRITELN(DATENROW2);
        WRITE(DATENROW2,≡*X   ≡);
        END;
    WRITELN(DATENROW1);
    WRITELN(DATENROW2);
    WRITELN(DATENRHS1,≡RHS≡);
    WRITELN(DATENRHS2,≡RHS≡);
    (*VERARBEITUNG PROBLEMDATEI-SAETZE TYP 2 *)
    RECCODE:=2;
    FOR I:=1 TO IMAXFZ DO
     BEGIN
     (*EINLESEN DATENSATZ*)
     IF ERRCODE = 0
        THEN BEGIN
           DATLES(DATENINPUT,RECCODE,RECLGE,ERRCODE);
           (*ABFRAGE,OB DATEI FEHLERFREI GELESEN*)
           IF ERRCODE = 0
             THEN BEGIN
                K:=DATENINPUT↑.SATZLAENGE;
                ZIELFKT(DATENINPUT,DATENROW1,DATENROW2,
                        DATENCOL1,DATENCOL2,K,I,RECCODE);
                 END;
        END;
   END;
    (*VERARBEITUNG PROBLEMDATEI-SAETZE TYP 3 *)
    RECCODE:=3;
    FOR I:=1 TO IMINFZ DO
     BEGIN
     (*EINLESEN DATENSATZ*)
     IF ERRCODE = 0
        THEN BEGIN
           DATLES(DATENINPUT,RECCODE,RECLGE,ERRCODE);
           (*ABFRAGE,OB DATEI FEHLERFREI GELESEN*)
           IF ERRCODE = 0
             THEN BEGIN
                K:=DATENINPUT↑.SATZLAENGE;
                ZIELFKT(DATENINPUT,DATENROW1,DATENROW2,
                        DATENCOL1,DATENCOL2,K,I,RECCODE);
                 END;
        END;
   END;
    (*VERARBEITUNG PROBLEMDATEI-SAETZE TYP 4 *)
    RECCODE:=4;
    FOR I:=1 TO IFRLE DO
     BEGIN
     (*EINLESEN DATENSATZ*)
     IF ERRCODE = 0
        THEN BEGIN
           DATLES(DATENINPUT,RECCODE,RECLGE,ERRCODE);
           (*ABFRAGE,OB DATEI FEHLERFREI GELESEN*)
           IF ERRCODE = 0
             THEN BEGIN
                K:=DATENINPUT↑.SATZLAENGE;
                FREST(DATENINPUT,DATENROW1,DATENROW2,
                        DATENCOL1,DATENCOL2,DATENRHS1,DATENRHS2,
                        K,I,RECCODE);
                 END;
        END;
END;
    (*VERARBEITUNG PROBLEMDATEI-SAETZE TYP 5 *)
    RECCODE:=5;
    FOR I:=1 TO IFRGE DO
     BEGIN
```

```
(*EINLESEN DATENSATZ*)
IF ERRCODE = 0
   THEN BEGIN
      DATLES(DATENINPUT,RECCODE,RECLGE,ERRCODE);
      (*ABFRAGE,OB DATEI FEHLERFREI GELESEN*)
      IF ERRCODE = 0
        THEN BEGIN
           K:=DATENINPUT↑.SATZLAENGE;
           FREST(DATENINPUT,DATENROW1,DATENROW2,
                 DATENCOL1,DATENCOL2,DATENRHS1,DATENRHS2,
                 K,I,RECCODE);
            END;
   END);
END);
(*VERARBEITUNG PROBLEMDATEI-SAETZE TYP 6 *)
RECCODE:=6;
FOR I:=1 TO IFRED DO
 BEGIN
 (*EINLESEN DATENSATZ*)
 IF ERRCODE = 0
   THEN BEGIN
      DATLES(DATENINPUT,RECCODE,RECLGE,ERRCODE);
      (*ABFRAGE,OB DATEI FEHLERFREI GELESEN*)
      IF ERRCODE = 0
        THEN BEGIN
           K:=DATENINPUT↑.SATZLAENGE;
           FREST(DATENINPUT,DATENROW1,DATENROW2,
                 DATENCOL1,DATENCOL2,DATENRHS1,DATENRHS2,
                 K,I,RECCODE);
            END;
   END);
END;
 (*VERARBEITUNG PROBLEMDATEI-SAETZE TYP 7 *)
 RECCODE:=7;
 FOR I:=1 TO ICRLE DO
  BEGIN
  (*EINLESEN DATENSATZ*)
  IF ERRCODE = 0
    THEN BEGIN
       DATLES(DATENINPUT,RECCODE,RECLGE,ERRCODE);
       (*ABFRAGE,OB DATEI FEHLERFREI GELESEN*)
       IF ERRCODE = 0
         THEN BEGIN
            K:=DATENINPUT↑.SATZLAENGE;
            CREST(DATENINPUT,DATENROW1,DATENROW2,
                  DATENCOL1,DATENCOL2,DATENRHS1,DATENRHS2,
                  K,I,RECCODE);
             END;
  END);
END;

 (*VERARBEITUNG PROBLEMDATEI-SAETZE TYP 8 *)
 RECCODE:=8;
 FOR I:=1 TO ICRGE DO
  BEGIN
  (*EINLESEN DATENSATZ*)
  IF ERRCODE = 0
    THEN BEGIN
       DATLES(DATENINPUT,RECCODE,RECLGE,ERRCODE);
       (*ABFRAGE,OB DATEI FEHLERFREI GELESEN*)
       IF ERRCODE = 0
         THEN BEGIN
            K:=DATENINPUT↑.SATZLAENGE;
            CREST(DATENINPUT,DATENROW1,DATENROW2,
                  DATENCOL1,DATENCOL2,DATENRHS1,DATENRHS2,
                  K,I,RECCODE);
             END;
```

```
                    END;
        END;
                (*VERARBEITUNG PROBLEMDATEI-SAETZE TYP 9 *)
                RECCODE:=9;
                FOR I:=1 TO NCREX DO
                 BEGIN
                 (*EINLESEN DATENSATZ*)
                 IF ERRCODE = 0
                    THEN BEGIN
                       DATLES(DATENINPUT,RECCODE,RECLGE,ERRCODE);
                       (*ABFRAGE,OB DATEI FEHLERFREI GELESEN*)
                       IF ERRCODE = 0
                        THEN BEGIN
                           K:=DATENINPUT↑.SATZLAENGE;
                           CREST(DATENINPUT,DATENROW1,DATENROW2,
                                  DATENCOL1,DATENCOL2,DATENRHS1,DATENRHS2,
                                  K,I,RECCODE);
                             END;
                    END;
        END;
    (* ABSCHLUSS DER DATEIEN *)
    WRITELN(DATENROW1,≡COLUMNS≡);
    WRITELN(DATENROW2,≡COLUMNS≡);
    WRITELN(DATENRHS1,≡ENDATA≡);
    WRITELN(DATENRHS2,≡ENDATA≡);
         END;
        END;
      END;
    END (* OF PROGRAM EINZELLP*).
```

END

FILMED

2-85

DTIC